

TUGAS AKHIR - KI141502

RANCANG BANGUN SIMULASI MENYALIP MOBIL (OVERTAKING CAR) MENGGUNAKAN STEERING WHEEL DAN OCULUS RIFT

FAJAR ADE PUTRA
NRP 5113100092

Dosen Pembimbing
Darlis Herumurti, S.Kom., M.Kom.
Ridho Rahman H., S.Kom., M.Sc.

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2017



TUGAS AKHIR - KI141502

RANCANG BANGUN SIMULASI MENYALIP MOBIL (OVERTAKING CAR) MENGGUNAKAN STEERING WHEEL DAN OCULUS RIFT

**FAJAR ADE PUTRA
NRP 5113100092**

**Dosen Pembimbing
Darlis Herumurti, S.Kom., M.Kom.
Ridho Rahman H., S.Kom., M.Sc.**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2017**



FINAL PROJECT- KI141502
IMPLEMENTATION VIRTUAL REALITY OF
OVERTAKING CAR USING STEERING WHEEL
AND OCULUS RIFT

FAJAR ADE PUTRA
NRP 5113100092

Advisor
Darlis Herumurti, S.Kom., M.Kom.
Ridho Rahman H., S.Kom., M.Sc.

DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY
SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY
SURABAYA
2017

LEMBAR PENGESAHAN

RANCANG BANGUN SIMULASI MENYALIP MOBIL (OVERTAKING CAR) MENGGUNAKAN STEERING WHEEL DAN OCULUS RIFT

Tugas Akhir

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Rumpun Mata Kuliah Interaksi Grafika dan Seni
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh:

FAJAR ADE PUTRA

NRP. 5113 100 092

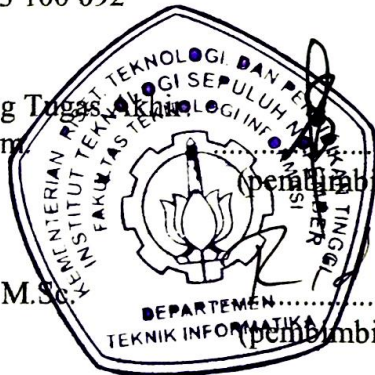
Disetujui oleh Dosen Pembimbing Tugas Akhir

Darlis Herumurti, S Kom., M Kom.

NIP: 19761215 200312 1 001

Ridho Rahman Hariadi, S Kom., M.Sc.

NIP: 19870213 201404 1 001



**SURABAYA
JUNI, 2017**

RANCANG BANGUN SIMULASI MENYALIP MOBIL (OVERTAKING CAR) MENGGUNAKAN STEERING WHEEL DAN OCULUS RIFT

Nama Mahasiswa : Fajar Ade Putra
NRP : 51 13 100 092
Jurusan : Teknik Informatika FTIf-ITS
Dosen Pembimbing I : Darlis Herumurti, S.Kom., M.Kom.
Dosen Pembimbing II : Ridho Rahman H., S.Kom., M.Sc.

ABSTRAK

Pada saat ini kendaraan bermotor sangat banyak digunakan dalam kehidupan sehari – hari dan kegiatan lainnya. Di Indonesia pertumbuhan perkembangan kendaraan bermotor sangat pesat, bahkan mencapai 10 persen pertahun. Dengan tingginya nilai perkembangan kendaraan mobil ini maka angka kecelakaan juga semakin meningkat. Penyebab kecelakaan itu, tidak hanya dari kesalahan mesin saja tapi juga dari kesalahan pengendara mobil sendiri. Salah satu penyebab kesalahan pengendara itu adalah karena kurang cakupannya seseorang dalam mengemudikan mobil untuk menyalip mobil.

Bagi para pengendara mobil seharusnya diberi informasi dan pembelajaran lebih agar mengetahui cara – cara dan tindakan yang harus dilakukan ketika mengalami kondisi – kondisi tertentu dalam berkendara. Oleh karena itu, salah satu solusi yang ditawarkan adalah dengan adanya aplikasi yang bisa digunakan untuk pembelajaran sekaligus bisa sebagai simulasi mengendarai mobil. Simulasi menyalip mobil dengan memanfaatkan teknologi Virtual Reality bisa menjadi salah satu solusi yang efektif untuk membiasakan para pengendara dalam menyalip mobil.

Pada tugas akhir ini uji coba dilakukan dengan metode kotak hitam dan kuisisioner dari pengguna. Kuisisioner ini menghasilkan nilai rata – rata 3.88 yang artinya aplikasi ini

sudah lebih dari cukup dan sesuai dengan keadaan aslinya. Dengan pengembangan lebih lanjut diharapkan aplikasi ini dapat membantu masyarakat untuk terbiasa dalam mengendarai mobil khususnya dalam kondisi menyalip mobil.

Kata kunci: Kecelakaan, Simulasi, Menyalip, Virtual Reality, Kotak Hitam, Kuisisioner.

IMPLEMENTATION VIRTUAL REALITY OF OVERTAKING CAR USING STEERING WHEEL AND OCULUS RIFT

Student Name : Fajar Ade Putra
NRP : 51 13 100 092
Major : Teknik Informatika FTIf-ITS
Advisor I : Darlis Herumurti, S.Kom., M.Kom.
Advisor II : Ridho Rahman H., S.Kom., M.Sc.

ABSTRACT

Currently, motor vehicles are very widely used in everyday life and other activities. In Indonesia the growth of motor vehicle growth is very rapid, even reaching 10 percent per year. With the high value of the development of this motor vehicle then the number of accidents is also increasing. The cause of the accident, not only from engine errors but also from motorist's own fault. One of the drivers' mistakes is the lack of driving skills to overtake a car.

For the driver of the car should be given more information and learn to know the ways and actions to be done when experiencing certain conditions in driving. Therefore, one of the solutions offered is the application that can be used for learning and simulation can drive a car. Simulating overtaking a car by utilizing Virtual Reality technology can be one effective solution to familiarize yourself with a motorist in overtaking a car.

In this final project experiment with black box method and questionnaire from user. This questionnaire yields an average value of 3.88 which means the application is more than adequate and in accordance with its original state. With further development is expected this application can help people to get used to driving a car, especially in overtaking the condition of the car.

Keywords: Accident, Simulation, Overtaking, Virtual Reality, Black Box.

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Segala puji dan syukur penulis hantarkan kehadiran tuhan yang maha esa Allah Subhanahu Wa Taa'la . Karena atas berkat dan rahmat beliau penulis bisa menyelesaikan Tugas Akhir yang merupakan salah satu syarat kelulusan dari Teknik Informatika ITS. Tidak lupa Salawat dan salam kepada Rasulullah Muhammad Sallallahu Alaihi Wasallam yang telah menghijrahka umat manusia dari jaman jahiliah.

Alhamdulillah pada kesempatan kali ini penulis bisa menyelesaikan salah satu kewajiban penulis untuk menyelesaikan perkuliahan di Teknik Informatika yaitu Tugas Akhir yang berjudul "Rancang Bangun Simulasi Menyalip Mobil (Overtaking Car) Menggunakan Steering Wheel dan oculus rift". Dalam pengerjaan tugas akhir ini merupakan momen untuk penulis untuk mengimplementasikan ilmu-ilmu yang didapat ketika masa perkuliahan di Teknik Informatika Institut Teknologi Sepuluh Nopember.

Melalui buku ini penulis juga berterima kasih kepada beberapa pihak yang telah membantu penulis dalam pengerjaan Tugas Akhir dan juga penyelesaian pembuatan buku ini baik yang terlihat langsung ataupun yang tidak terlihat langsung. Pihak-pihak tersebut adalah:

1. Allah SWT yang telah melimpahkan rahmat, hidayah, dan kemudahaNya. Sehingga penulis mampu menyelesaikan tugas akhir dengan baik pada waktunya.
2. Rasullullah Nabi Muhammad SAW yang telah membawa umatnya dari zaman kekufuran ke zaman yang baik seperti saat ini dan juga menjadi inspirasi contoh yang baik bagi penulis dan umatnya.
3. Kedua orang tua Muzuardi (Ayah), Sulasteti (Ibu), serta kedua orang saudara yaitu Nelvi Yanti (Kakak) dan Zulhuda (Abang) dan keluarga yang selalu memberikan dukungan dan semangat baik dalam bentuk motivasi

ataupun materi yang membuat penulis menjadi semangat dalam mengerjakan Tugas Akhir.

4. Bapak Darlis Herumurti, S. Kom., M. Kom. dan Bapak Ridho Rahman H., S.Kom., M.Sc. yang telah bersedia menjadi dosen pembimbing tugas akhir, sehingga penulis dapat mengerjakan tugas akhir dengan arahan dan bimbingan dari beliau.
5. Teman-teman Mahasiswa Teknik Informatika Angkatan 2013 yang telah bersama-sama berjuang menempuh pendidikan di departemen Teknik Informatika.
6. Teman – teman satu daerah yang terus memberi semangat dukungan dan juga teman – teman dalam menghabiskan hari selama di kuliah di surabaya.
7. Teman - teman yang telah membantu dan menemani penulis dalam penyusunan Tugas Akhir ini diantaranya Cayza, Naufal, dan juga Mas Dimas Widdy yang bersedia direpotkan untuk ditanya – tanyai mengenai tugas akhir ini.
8. Teman – teman lab IGS (Interaksi Grafika dan Seni) dari angkatan 13, 14, 15 yang saling mendukung dalam mengerjakan tugas akhir ini dan juga yang menghibur penulis dalam keadaan suntuk.
9. Teman-teman seperjuangan di HMTTC Optimasi yang saling menyemangati satu sama lain dan teman2 dan staff Medfo 4th Gen yang mengingatkan dan menjadi teman baik sehari-harinya.

Penulis telah berusaha dengan maksimal dalam perancangan aplikasi dan buku ini. Apabila ada kesalahan, kekurangan dan kelainan dalam buku dan aplikasi ini penulis mohon maaf. Kritik dan saran sangat penulis harapkan dalam membangun aplikasi ini lebih baik kedepanya.

Surabaya, Juni 2017

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN.....	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR.....	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xv
DAFTAR TABEL	xvii
DAFTAR KODE SUMBER	xix
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	1
1.3 Batasan Masalah	2
1.4 Tujuan.....	2
1.5 Manfaat.....	2
1.6 Metodologi	2
1.7 Sistematika Penulisan	4
BAB II TINJAUAN PUSTAKA.....	7
2.1 Realitas Virtual	7
2.2 Unity	8
2.3 Steering Wheel	8
2.4 Menyalip Mobil Berdasarkan PP 1993.....	9
2.5 Oculus Rift.....	11
2.6 Oculus Utility	12
BAB III ANALISIS DAN PERANCANGAN	13
3.1 Analisis Perangkat Lunak.....	13
3.1.1 Deskripsi Umum Perangkat Lunak.....	13
3.1.2 Spesifikasi Kebutuhan Perangkat Lunak	14
3.1.3 Identifikasi Pengguna	15
3.2 Perancangan Perangkat Lunak.....	15
3.2.1 Perancangan Diagram Kasus Penggunaan.....	16
3.2.2 Definisi Kasus Penggunaan	17
3.2.3 Definisi Aktor	20

3.2.4	Perancangan Model.....	20
3.2.5	Arsitektur Umum Aplikasi.....	23
3.2.6	Perancangan Antarmuka Aplikasi.....	23
3.2.7	Perancangan Skenario Simulasi Menyalip Mobil	25
	BAB IV IMPLEMENTASI.....	29
4.1	Lingkungan Implementasi	29
4.2	Implementasi Antarmuka.....	29
4.2.1	Implementasi Antarmuka Halaman Menu Utama....	29
4.2.2	Implementasi Antarmuka Halaman Simulasi	33
4.3	Implementasi Integrasi Oculus Dengan Unity.....	45
4.4	Implementasi Sistem Kerja Mobil	47
4.5	Implementasi Skenario Simulasi Menyalip Mobil	52
4.5.1	Implementasi Skenario Level 1.....	53
4.5.2	Implementasi Skenario Level 2.....	54
4.5.3	Implementasi Skenario Level 3.....	55
4.5.4	Implementasi Mobil yang akan Disalip	56
4.5.5	Implementasi Random Posisi Mobil	58
	BAB V PENGUJIAN DAN EVALUASI	63
5.1	Lingkungan Uji Coba	63
5.2	Skenario Pengujian	63
5.2.1	Pengujian Fungsionalitas	64
5.2.2	Pengujian Aplikasi Terhadap Pengguna	70
5.3	Evaluasi	71
5.3.1	Evaluasi Pengujian Fungsionalitas.....	71
5.3.2	Evaluasi Pengujian Aplikasi Terhadap Pengguna....	72
	BAB VI KESIMPULAN DAN SARAN.....	75
6.1.	Kesimpulan	75
6.2.	Saran	76
	DAFTAR PUSTAKA	77
	LAMPIRAN	79
	BIODATA PENULIS	87

DAFTAR GAMBAR

Gambar 2.1 Contoh Tampilan realitas Virtual	7
Gambar 2.2 Genius Speed Wheel 6MT.....	9
Gambar 2.3 Oculus Rift.....	12
Gambar 3.1 Diagram Kasus Penggunaan	16
Gambar 3.2 Perancangan Model 3D Mobil Tampak Dalam	20
Gambar 3.3 Perancangan Arsitektur Aplikasi	23
Gambar 3.4 Rancangan Antarmuka Menu Utama	24
Gambar 3.5 Rancangan Antarmuka Simulasi Menyalip Mobil ..	25
Gambar 3.6 Rancangan Skenario Level 1	26
Gambar 3.7 Rancangan Skenario Level 2	27
Gambar 3.8 Rancangan Skenario Level 3	28
Gambar 4.1 Antarmuka Halaman Menu Utama.....	30
Gambar 4.2 Pembuatan Input Navigation	32
Gambar 4.3 Perubahan Pada EventSystem	33
Gambar 4.4 Antarmuka Halaman Simulasi.....	34
Gambar 4.5 Tampilan steer pada mobil	35
Gambar 4.6 Tampilan Skor Menyalip.....	36
Gambar 4.7 Tampilan Kaca Spion Tengah	36
Gambar 4.8 Pembuatan Texture Render Kaca Spion Tengah	37
Gambar 4.9 Pembuatan Kamera Kaca Spion Tengah	38
Gambar 4.10 Pembuatan Raw Image Kaca Spion Tengah.....	38
Gambar 4.11 Tampilan Kaca Spion Kanan.....	39
Gambar 4.12 Pembuatan Texture Render Kaca Spion Kanan.....	40
Gambar 4.13 Pembuatan Kamera Kaca Spion Kanan.....	40
Gambar 4.14 Pembuatan Raw Image Kaca Spion Kanan	41
Gambar 4.15 Tampilan Kaca Spion Kiri.....	41
Gambar 4.16 Pembuatan Terxure Render Kaca Spion Kiri	42
Gambar 4.17 Pembuatan Kamera Kaca Spion Kiri.....	43
Gambar 4.18 Pembuatan Raw Image Kaca Spion Kiri	43
Gambar 4.19 Penambah Input Untuk Perubahan Gear	44
Gambar 4.20 Tampilan Direktori yang Digunakan.....	46
Gambar 4.21 Tampilan Player Setting	46

Gambar 4.22 Tampilan Box dan Wheel Collider	48
Gambar 4.23 Implementasi Skenario Level 1	53
Gambar 4.24 Implementasi Skenario level 2.....	54
Gambar 4.25 Implementasi Skenario Level 3	55
Gambar 4.26 Implementasi Mobil yang akan Disalip	56
Gambar 4.27 Implementasi Node – node rute mobil.....	59
Gambar 5.1 Kondisi Awal dan Akhir Pengujian Memilih Level 64	
Gambar 5.2 Kondisi Awal dan Akhir Pengujian Secara Gagal...66	
Gambar 5.3 Kondisi Awal dan Akhir Pengujian Level 1	67
Gambar 5.4 Kondisi Awal dan Akhir Pengujian Level 2.....	68
Gambar 5.5 Kondisi Awal dan Akhir Pengujian Level 3	69

DAFTAR TABEL

Tabel 3.1 Karakteristik Pengguna	15
Tabel 3.2 Skenario Kasus Penggunaan	16
Tabel 3.3 Skenario Kasus Pengguna Memilih Level Simulasi ...	17
Tabel 3.4 Skenario Kasus Menjalankan Simulasi Menyalip	18
Tabel 3.5 Deskripsi pengguna	20
Tabel 3.6 Daftar Asset	21
Tabel 4.1 Lingkungan Implementasi Perangkat Keras & Lunak	29
Tabel 5.1 Lingkungan Pengujian Perangkat Keras dan Lunak ...	63
Tabel 5.2 Pengujian Memilih Level Simulasi	65
Tabel 5.3 Tabel Pengujian Menjalankan Simulasi Secara Gagal	66
Tabel 5.4 Tabel Pengujian Menjalankan Level 1	67
Tabel 5.5 Tabel Pengujian Menjalankan Level 2	68
Tabel 5.6 Tabel Pengujian Menjalankan Level 3	69
Tabel 5.7 Kuesioner Pengguna Simulasi Menyalip Mobil	70
Tabel 5.8 Tabel Hasil Evaluasi Pengujian Fungsionalitas	71
Tabel 5.9 Hasil Kuesioner Pengguna	72

DAFTAR KODE SUMBER

Kode Sumber 4.1 Kode Sumber Antarmuka Menu Utama	31
Kode Sumber 4.2 Kode Sumber steer pada mobil	35
Kode Sumber 4.3 Kode Sumber Perubahan Teks Gear.....	45
Kode Sumber 4.4 Kode Sumber Sistem Kerja Mobil	52
Kode Sumber 4.5 Kode Sumber Mobil yang Disalip.....	58
Kode Sumber 4.6 Kode Sumber Mengabungkan Node – node...	61
Kode Sumber 4.7 Kode Sumber Random Posisi Mobil	61

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kecelakaan lalu lintas sudah menjadi momok yang sangat menakutkan bagi bangsa Indonesia, terlebih di Indonesia ada musim mudik yang membuat arus mudik meningkat pesat setiap tahunnya. Orang-orang secara serentak dan beramai-ramai pulang ke kampung halaman dengan tujuan silaturahmi dengan keluarga masing-masing. Dengan Intensitas yang padat tersebut maka peluang kecelakaan mobil sangat besar kemungkinan terjadi dengan beberapa faktor diantaranya akibat kesalahan pengemudi mobil itu sendiri.

Dikutip dari salah satu lembaga kesehatan dunia yang berada dibawah naungan PBB (WHO) yang merilis daftar angka kecelakaan lalu lintas di 180 negara pada tahun 2015. Indonesia berada di posisi 3 di daftar negara dengan angka kematian terbesar akibat kecelakaan dibawah China dan India dengan total 38.279 kematian. Dari statistik beberapa tahun belakangan ini Indonesia selalu mengalami peningkatan angka kecelakaan setiap tahunnya.

Simulasi yang dibuat ini memiliki tujuan agar calon pengemudi merasakan mengendarai mobil seperti pada umumnya. Sehingga pengemudi terbiasa dalam mengendarai mobil dan juga merasakan menyalip mobil dengan baik dan benar. Dengan menggunakan *Steering Wheel* dan Oculus Rift, pengemudi diharapkan dapat merasakan suasana ujian yang nyata tanpa adanya rasa khawatir akan insiden – insiden yang mungkin terjadi dalam mengendarai mobil.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam Tugas Akhir ini adalah sebagai berikut:

1. Bagaimana cara menyalip mobil yang benar?
2. Bagaimana cara membuat simulasi mengendarai mobil dengan steering wheel di Unity3D?

3. Bagaimana mengkoneksikan steering wheel dan oculus rift di Unity3D?

1.3 Batasan Masalah

Permasalahan yang dibahas dalam Tugas Akhir ini memiliki beberapa batasan, di antaranya sebagai berikut:

1. Menggunakan *Steering Wheel* Genius Speed Wheel 6 MT.
2. Perangkat VR yang di pakai adalah Oculus Rift Development kit 2 (DK2).
3. Platform yang di gunakan adalah Unity3D.
4. Simulasi menyalip mobil ini hanya untuk jalan yang menggunakan dua jalur.
5. Peraturan menyalip mobil sesuai dengan Peraturan Pemerintah No. 43/1993 pasal 52 - 55

1.4 Tujuan

Tujuan dari pembuatan Tugas Akhir ini antara lain:

1. Pengemudi mengetahui cara menyalip mobil di jalanan dua jalur dengan benar.
2. Melatih para pengemudi dengan rasa aman dan tanpa rasa khawatir akan insiden – insiden dalam berkendara.
3. Untuk dapat mengintegrasikan *Steering Wheel* dan Oculus Rift dengan Unity3D.

1.5 Manfaat

Manfaat dari hasil pembuatan Tugas Akhir ini adalah sebagai sarana berlatih untuk para pengemudi mobil agar bisa menyalip mobil dengan benar.

1.6 Metodologi

Pembuatan Tugas Akhir dilakukan menggunakan metodologi sebagai berikut:

A. Studi literatur

Tahap studi literatur merupakan tahap pembelajaran dan pengumpulan informasi yang digunakan untuk mengimplementasikan Tugas Akhir. Tahap ini diawali dengan pengumpulan literatur, diskusi, eksplorasi teknologi dan pustaka, serta pemahaman dasar teori yang digunakan pada topik Tugas Akhir. Literatur-literatur yang dimaksud adalah sebagai berikut:

1. Realitas Virtual
2. Unity3D
3. Menyalip Mobil Berdasarkan PP 1993
4. *Steering Wheel*
5. Oculus Rift

B. Perancangan perangkat lunak

Pada tahap ini diawali dengan melakukan analisis awal terhadap permasalahan utama yang muncul pada topik Tugas Akhir. Kemudian dilakukan perancangan perangkat lunak yang meliputi penentuan data yang digunakan dan proses-proses yang akan dilaksanakan. Langkah yang digunakan pada tahap ini adalah sebagai berikut:

1. Pencarian dan pendataan tentang menyalip mobil meliputi objek-objek apa saja yang dibuat dalam bentuk 3D.
2. Perancangan skenario ujian yang akan disimulasikan.
3. Perancangan visualisasi untuk membangun aplikasi realitas nyata.
4. Perancangan integrasi aplikasi dengan perangkat Oculus Rift.

C. Implementasi dan pembuatan sistem

Pada tahap ini dilakukan implementasi pemodelan objek-objek 3D untuk realitas virtual. Kemudian dilakukan integrasi

aplikasi dengan perangkat Oculus Rift. Aplikasi ini dibangun dengan Unity3D.

D. Uji coba dan evaluasi

Pada tahap ini dilakukan uji coba dengan menggunakan beberapa macam kondisi untuk mencoba aplikasi bisa berjalan atau tidak. Uji fungsionalitas untuk mengetahui apakah aplikasi sudah memenuhi semua kebutuhan fungsional.

E. Penyusunan laporan Tugas Akhir

Pada tahap ini dilakukan penyusunan laporan yang berisi dasar teori, dokumentasi dari perangkat lunak, dan hasil-hasil yang diperoleh selama pengerjaan Tugas Akhir.

1.7 Sistematika Penulisan

Buku Tugas Akhir ini terdiri dari beberapa bab, yang dijelaskan sebagai berikut:

BAB I PENDAHULUAN

Bab ini berisi latar belakang masalah, rumusan dan batasan permasalahan, tujuan dan manfaat pembuatan Tugas Akhir, metodologi yang digunakan, dan sistematika penyusunan Tugas Akhir.

BAB II TINJAUAN PUSTAKA

Bab ini membahas dasar pembuatan dan beberapa teori penunjang yang berhubungan dengan pokok pembahasan yang mendasari pembuatan Tugas Akhir ini.

BAB III ANALISIS DAN PERANCANGAN

Bab ini membahas analisis dari sistem yang dibuat meliputi analisis permasalahan, deskripsi umum perangkat lunak, spesifikasi kebutuhan, dan identifikasi pengguna. Kemudian membahas rancangan dari sistem yang dibuat meliputi rancangan skenario kasus penggunaan, arsitektur, data, dan antarmuka.

BAB IV IMPLEMENTASI

Bab ini membahas implementasi dari rancangan sistem yang dilakukan pada tahap perancangan. Penjelasan implementasi meliputi implementasi pembuatan realitas virtual ibadah haji dengan menggunakan Oculus Rift, dan antarmuka aplikasi.

BAB V PENGUJIAN DAN EVALUASI

Bab ini membahas pengujian dari aplikasi yang dibuat dengan melihat keluaran yang dihasilkan oleh aplikasi dan evaluasi untuk mengetahui kemampuan aplikasi.

BAB VI PENUTUP

Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan serta saran untuk pengembangan aplikasi selanjutnya.

BAB II

TINJAUAN PUSTAKA

2.1 Realitas Virtual

Realitas virtual adalah sebuah teknologi dimana pengguna seolah – olah dapat merasakan berada di lingkungan yang telah disediakan. Dengan menggunakan realitas virtual, pengguna dapat berinteraksi secara tidak langsung ke dalam suasana tiga dimensi yang seolah nyata. Kebanyakan virtual realitas sekarang ini menggunakan alat penampil stereoskopik sebagai penampil visual. Karena realitas virtual ini adalah representatif dari lingkungan tiga dimensi yang tidak nyata, maka komputasi komputer dan visualisasi menjadi kunci kualitas realitas virtual itu sendiri [1].

Realitas virtual berbeda dengan animasi ataupun video yang memiliki skenario tetap yang apabila diulangi akan berasa sama dengan yang sebelumnya. Dengan realitas virtual, pengguna dapat melakukan berbagai interaksi dan dapat melihat dari berbagai perspektif. Realitas virtual ini memberikan fleksibilitas kepada penggunanya agar dapat menjelajah lingkungan virtual yang tidak nyata tersebut sesuai dengan keinginannya walaupun terbatas. Gambar 2.1 adalah contoh bentuk tampilan realitas virtual apabila ditampilkan di komputer.



Gambar 2.1 Contoh Tampilan realitas Virtual

2.2 Unity

Unity adalah sebuah tool pengembangan game yang terintegrasi kuat dengan satu set lengkap alat intuitif dan alur kerja yang cepat untuk membuat 3D interaktif dan konten 2D. Unity merupakan *easy multiplatform publishing*. Unity memiliki *Asset Store* yang menyediakan *Asset* untuk di unduh secara gratis maupun berbayar. Terdapat pula *Unity Community* yang menyediakan tutorial secara gratis untuk semua pengguna unity dalam Bahasa inggris. Fitur scripting yang disediakan, mendukung 3 bahasa pemrograman, yaitu JavaScript, C#, dan Boo. *Flexible and EasyMoving, rotating, dan scaling objects* hanya perlu sebaris kode. Begitu juga dengan *Duplicating, removing, dan changing properties*. Visual Properties Variables yang di definisikan dengan scripts ditampilkan pada Editor. Bisa digeser, di *drag and drop*, bisa memilih warna dengan *color picker*. Berbasis .NET. Artinya perjalanan program dilakukan dengan *Open Source .NET platform, Mono*. Serta mendukung pengembangan aplikasi Microsoft, SONY, Qualcomm, BlackBerry, Samsung, Ninetendo, Oculus VR dan Intel [5].

2.3 Steering Wheel

Steering Wheel adalah suatu input untuk merealisasikan *game control*. Sesuai dengan namanya *game control* adalah perangkat *plug-in* yang di hubungkan sebuah perangkat sehingga memudahkan untuk berinteraksi ke dalam game tersebut. Contoh lain dari *game control* ini adalah *keyboard, mouse, gamepad, joystick*, dan lain sebagainya. Karena simulasi yang akan dibuat ini berbasis *driving simulator*, maka akan digunakan *steering wheel* Genius Speed Wheel 6 MT sebagai sebuah game controlnya. Agar calon pengemudi dapat merasakan seperti menggunakan mobil seperti aslinya. Gambar 2.2 merupakan contoh tampilan dari *steering wheel*.



Gambar 2.2 Genius Speed Wheel 6MT

Genius Speed Wheel 6MT merupakan jenis *steering wheel* yang memiliki bagian 2 bagian yaitu bagian atas dan bawah. Pada bagian atas terdapat setir yang digunakan untuk mengendalikan mobil ke kanan dan kiri. Terdapat juga tuas yang digunakan untuk mengganti gigi transmisi pada mobil. Pada bagian bawah terdapat dua buah pedal yang berfungsi untuk menaikkan *throttle* pada mobil dan mengerem mobil. Sangat disayangkan *steer* yang terdapat pada *steering wheel* ini hanya bisa melakukan putaran ke kanan dan ke kiri sebanyak 90°. Tidak seperti mobil sesungguhnya yang *steernya* dapat berputar sebanyak 360°.

2.4 Menyalip Mobil Berdasarkan PP 1993

Kecelakaan lalu lintas di Indonesia menjadi momok yang menakutkan, Karena selalu menelan korban jiwa yang banyak. Kecelakaan lalu lintas ini tidak hanya di Indonesia saja yang menakutkan, tapi hampir semua negara di dunia ini. Untuk Indonesia sendiri, angka kecelakaan setiap tahunnya terus bertambah,

berbanding lurus dengan angka penjualan kendaraan yang juga mengalami kenaikan setiap tahunnya.

Untuk menyalip mobil sendiri, telah ada peraturan yang mengaturnya, yaitu Peraturan pemerintah No. 43/1993 pasal 52 – 55. Dalam peraturan pemerintah tersebut telah dijelaskan bagaimana ketentuan – ketentuan pengemudi yang akan menyalip mobil. Pasal – pasal tersebut berbunyi:

- Pasal 52

- 1) Pengemudi yang akan melewati kendaraan lain harus mempunyai pandangan bebas dan menjaga ruang yang cukup bagi kendaraan yang dilewatinya.
- 2) Pengemudi sebagaimana dimaksud dalam ayat (1), mengambil lajur atau jalur jalan sebelah kanan dari kendaraan yang dilewati.
- 3) Dalam keadaan tertentu pengemudi sebagaimana dimaksud dalam ayat (2) dapat mengambil lajur atau jalur jalan sebelah kiri dengan tetap memperhatikan keselamatan lalu lintas.
- 4) Keadaan tertentu sebagaimana dimaksud dalam ayat (3) meliputi:
 - a. lajur sebelah kanan atau lajur paling kanan dalam keadaan macet;
 - b. bermaksud akan belok kiri.
- 5) Apabila kendaraan yang akan dilewati telah memberi isyarat akan mengambil lajur atau jalur jalan sebelah kanan, pengemudi sebagaimana dimaksud dalam ayat (1), pada saat yang bersamaan dilarang melewati kendaraan tersebut.

- Pasal 53

Pengemudi harus memperlambat kendaraannya apabila akan melewati:

- a. kendaraan umum yang sedang berada pada tempat turun- naik penumpang;
 - b. kendaraan tidak bermotor yang ditarik oleh hewan, hewan yang ditunggangi, atau hewan yang digiring.
- Pasal 54
 - 1) Pengemudi mobil bus sekolah yang sedang berhenti untuk menurunkan dan atau menaikkan anak sekolah wajib menyalakan tanda lampu berhenti mobil bus sekolah.
 - 2) Pengemudi kendaraan yang berada di belakang mobil bus sekolah yang sedang berhenti sebagaimana dimaksud dalam ayat (1) wajib menghentikan kendaraannya.
- Pasal 55

Pengemudi dilarang melewati:

 - a. kendaraan lain di persimpangan atau persilangan sebidang;
 - b. kendaraan lain yang sedang memberi kesempatan menyeberang kepada pejalan kaki atau pengendara sepeda.

2.5 Oculus Rift

Oculus Rift merupakan sebuah perangkat yang ditemukan oleh Palmer luckey yang juga dikenal dengan pendiri Oculus VR. Oculus Rift digunakan untuk memanipulasi objek – objek dalam dunia maya terlihat nyata. Oculus Rift sendiri adalah sebuah kaca mata VR yang mempunyai dua kaca mata yang diperuntukan untuk kedua mata masing-masing yang memungkinkan untuk melihat keberbagai arah, dengan sudut pandang horizontal mencapai 90 derajat dan sudut pandang diagonal mencapai 110 derajat. Oculus sendiri telah merilis

Oculus Rift untuk dijual yaitu Developer Kit 1 (DK1) dan Developer Kit 2 (DK2). Gambar 2.3 adalah contoh gambar dari Oculus Rift.



Gambar 2.3 Oculus Rift

2.6 Oculus Utility

Oculus utility adalah sebuah *package* yang apabila dibuka terdapat beberapa *asset* unity yang akan langsung masuk ke *project* yang sedang dibuka. *Asset* yang ada ada oculus utility ini berhubungan dengan pengembangan menggunakan oculus pada unity. Terdapat *asset* yang penting contohnya adalah OVRCameraRig. Ini adalah sebuah kamera pada unity yang memang dikhususkan untuk pengguna oculus. Jarak pandang yang sudah diatur dan beberapa fitur memang ditujukan lebih untuk aplikasi berbasis oculus.

BAB III

ANALISIS DAN PERANCANGAN

Bab ini menjelaskan tentang analisis dan perancangan simulasi menyalip mobil menggunakan *Steering* dan *Oculus Rift* . Pembahasan yang akan dilakukan meliputi analisis fitur yang dibutuhkan dan perancangan perangkat lunak.

3.1 Analisis Perangkat Lunak

Subbab ini menjelaskan tentang hasil analisis kebutuhan perangkat lunak serta arsitektur simulasi menyalip mobil menggunakan *Steering Wheel* dan *Oculus Rift*. Tiap-tiap subbab menjelaskan tentang deskripsi umum perangkat lunak, spesifikasi kebutuhan perangkat lunak, analisis aktor, arsitektur perangkat lunak, dan skenario kasus penggunaan.

3.1.1 Deskripsi Umum Perangkat Lunak

Pada Tugas Akhir ini dibuat sebuah simulasi menyalip mobil menggunakan *Steering Wheel* dan *Oculus Rift*. Dikarenakan angka kecelakaan mobil yang terus bertambah setiap tahunnya, maka dirasa perlu untuk memberikan pencerdasan tentang bagaimana cara menyalip mobil sesuai dengan peraturan yang berlaku yaitu dengan aplikasi menyalip mobil ini. Agar simulasi ini seperti simulasi membawa mobil sebenarnya maka digunakan *Steering Wheel* supaya user merasakan suasana mengemudi seperti aslinya. Dan untuk memberikan simulasi visual yang nyata maka digunakan *Oculus* sebagai toolnya.

Simulasi ini merupakan simulasi menyalip mobil yang bertujuan untuk membantu orang supaya terbiasa dalam menyalip mobil supaya terhindar dari kecelakaan. Untuk membuat terbiasa menyalip mobil dalam beberapa kondisi, maka dalam simulasi ini dibagi menjadi 3 tingkatan kesulitan dari yang mudah, sedang dan susah.

Pengguna akan diminta untuk memilih tingkatan kesulitan yang ingin dicoba. Setelah memilih tingkat kesulitan, maka

pengguna akan berada disebuah mobil dan siap untuk dijalankan. Mobil tersebut akan dilengkapi dengan kaca spion kanan, kiri dan tengah. Apabila pengguna menabrak trotoar atau menabrak mobil lain maka simulasi akan langsung gagal dan harus mengulang kembali dari awal. Apabila pengguna berhasil menyelesaikan ujian maka akan dilanjutkan dengan ujian dengan tingkatan yang lebih sulit. Dan apabila berhasil menyelesaikan level yang paling sulit maka akan kembali ke menu utama.

3.1.2 Spesifikasi Kebutuhan Perangkat Lunak

Kebutuhan sistem yang dibuat ini melibatkan dua hal, yaitu kebutuhan fungsional dan non-fungsional. Dua kebutuhan tersebut akan dijelaskan lebih lanjut pada subbab berikut ini.

3.1.2.1 Kebutuhan Fungsional Perangkat Lunak

Berdasarkan deskripsi umum aplikasi, maka disimpulkan bahwa kebutuhan fungsional dari aplikasi ini adalah sebagai berikut.

1. Memilih tingkat kesulitan menyalip mobil

Aplikasi dapat memiliki fungsionalitas memilih tingkatan ujian yang ingin dicoba. Pengguna harus menyelesaikan ujian 1 untuk dapat mencoba ujian 2 dan seterusnya. Tingkatan pada aplikasi ini sendiri terbagi menjadi tiga. Terdapat ujian dari tingkatan mudah, sedang dan ujian yang paling susah. Setelah ujian tingkatan yang mudah diselesaikan maka akan dilanjutkan dengan ujian dengan tingkatan yang lebih sulit.

2. Menjalankan simulasi menyalip mobil

Aplikasi dapat menampilkan *scene* dimana pengguna dapat menggunakan mobil yang seperti sesungguhnya. pengguna dapat mengendarainya dan melakukan serangkaian tingkat kesulitan yang di pilihnya dari tingkatan mudah, sedang dan tingkatan yang paling sulit.

3.1.2.2 Kebutuhan Non-Fungsional Perangkat Lunak

Terdapat beberapa kebutuhan non-fungsional yang apabila dipenuhi, dapat meningkatkan kualitas dari aplikasi ini. Berikut daftar kebutuhan non-fungsional:

1. Kebutuhan grafis

Grafis yang bagus dapat menambahkan nilai yang lebih terhadap suatu aplikasi. Kemiripan suatu objek pada aplikasi ini dengan nyatanya dapat membuat pengguna memiliki pengalaman yang lebih maksimal simulasi.

2. Menciptakan suasana yang mirip dengan sesungguhnya

Pengguna akan mendapatkan pengalaman yang maksimal apabila dapat merasakan hal yang mirip dengan sesungguhnya. Dalam konteks ini adalah suasana membawa mobil dan menyalip mobil.

3.1.3 Identifikasi Pengguna

Berdasarkan deskripsi umum diatas, maka dapat diketahui bahwa pengguna yang akan menggunakan aplikasi ini hanya satu orang, yaitu pengguna yang menjalankan aplikasi. Karakteristik pengguna tercantum dalam Tabel 3.1.

Tabel 3.1 Karakteristik Pengguna

Nama Aktor	Tugas	Hak Akses Aplikasi	Kemampuan yang harus dimiliki
Pengguna	Pihak luar yang mencoba aplikasi	Menjalankan aplikasi	Tidak ada

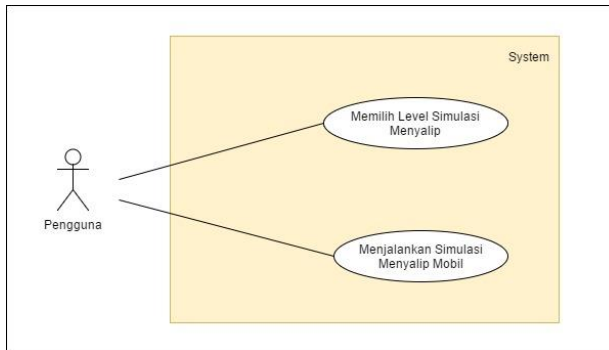
3.2 Perancangan Perangkat Lunak

Tahap perancangan dalam subbab ini dibagi menjadi beberapa bagian yaitu perancangan diagram kasus penggunaan, perancangan skenario kasus penggunaan, perancangan data, perancangan model,

perancangan informasi, perancangan antarmuka pengguna, dan perancangan kontrol aplikasi.

3.2.1 Perancangan Diagram Kasus Penggunaan

Dalam aplikasi Tugas Akhir ini, terdapat dua kasus penggunaan yang ada yaitu memilih tingkatan kesulitan dan menjalankan aplikasi. Rancangan scenario kasus penggunaan dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram Kasus Penggunaan

Penjelasan singkat dari masing-masing kasus penggunaan dapat dilihat pada Tabel 3.2.

Tabel 3.2 Skenario Kasus Penggunaan

No	Kode Kasus Penggunaan	Nama Kasus Penggunaan	Keterangan
1	UC-001	Memilih level simulasi menyalip mobil	Pengguna memilih tingkatan kesulitan yang akan dijalankan di aplikasi.
2	UC-002	Menjalankan simulasi menyalip mobil	Pengguna menjalankan simulasi menyalip mobil dengan aplikasi.

3.2.2 Definisi Kasus Penggunaan

Subbab menjelaskan kasus penggunaan. Detail mengenai kasus penggunaan tersebut dapat dilihat pada subbab berikut ini.

3.2.2.1 Memilih level kesulitan simulasi menyalip

Dalam kasus ini pengguna diharapkan memilih satu dari tiga skenario simulasi menyalip yang tersedia. Sehingga pengguna dapat melatih kemampuannya dalam menyalip mobil. Spesifikasi kasus penggunaan memilih jenis ujian dapat dilihat pada Tabel 3.3.

Tabel 3.3 Skenario Kasus Pengguna Memilih Level Simulasi

Nama Kasus Penggunaan	Memilih level simulasi menyalip mobil
Kode	UC-001
Deskripsi	Kasus penggunaan agar actor dapat memilih level simulasi menyalip.
Aktor	Pengguna
Kondisi Awal	Pengguna sudah masuk ke aplikasi, dan sudah menggunakan oculus rift dan <i>steering wheel</i> .
Alur Kejadian Normal	<ol style="list-style-type: none"> 1. Aplikasi akan menampilkan button pilihan level simulasi menyalip. 2. Pengguna menggerakkan steer pada <i>steering wheel</i> ke kanan untuk menggerakkan hover tombol ke ujian berikutnya. <ol style="list-style-type: none"> A1. Pengguna menggerakkan steer pada <i>steering wheel</i> ke kiri untuk menggerakkan hover tombol ke ujian sebelumnya. 3. Pengguna memilih ujian yang ingin dijalankan. 4. Pengguna menekan tombol x pada <i>steering wheel</i> sebagai tombol pilih. 5. Aplikasi menampilkan <i>scene</i> ujian sim yang di pilih

Nama Kasus Penggunaan	Memilih level simulasi menyalip mobil
Alur Kejadian Alternatif	<p>A1. Pengguna menggerakkan steer pada <i>steering wheel</i> ke kiri untuk menggerakkan hover ke ujian sebelumnya</p> <ol style="list-style-type: none"> 1. Steer pada <i>steering wheel</i> digerakkan ke kiri untuk menggerakkan hover tombol ke ujian sebelumnya 2. Kembali ke alur kejadian normal nomor 3

3.2.2.2 Menjalankan simulasi menyalip mobil

Dalam kasus ini pengguna diharapkan dapat melakukan simulasi skenario ujian yang telah dipilih. Terdapat tiga tingkatan simulasi yaitu mudah, sedang dan sulit. Spesifikasi kasus penggunaan menjalankan simulasi menyalip mobil dapat dilihat pada Tabel 3.4.

Tabel 3.4 Skenario Kasus Menjalankan Simulasi Menyalip

Nama Kasus Penggunaan	Menjalankan simulasi menyalip mobil
Kode	UC-002
Deskripsi	Kasus penggunaan agar aktor dapat menjalankan Simulasi Menyalip
Aktor	Pengguna
Kondisi Awal	Pengguna sudah masuk ke salah satu skenario simulasi menyalip mobil, dan sudah memakai oculus rift dan <i>steering wheel</i> .

Alur Normal	<ol style="list-style-type: none"> 1. Aplikasi <i>scene</i> dengan keadaan mengendari mobil. 2. Aplikasi menampilkan skor yang akan dicapai. 3. Pengguna melihat skor yang ingin dicapai. 4. Pengguna menginjak pedal kanan pada <i>steering wheel</i>. <ol style="list-style-type: none"> A1. Pengguna menginjak pedal kiri pada <i>steering wheel</i>. A2. Pengguna menggerakkan <i>steer</i> pada <i>steering wheel</i> ke kanan. A3. Pengguna menggerakkan <i>steer</i> pada <i>steering wheel</i> ke kiri. 5. Mobil yang dikendarai pengguna pada aplikasi menambah kecepatan. 6. Pengguna berhasil menjalankan skenario pada simulasi. <ol style="list-style-type: none"> A4. Pengguna gagal menjalankan skenario pada simulasi.
Alur Alternatif	<ol style="list-style-type: none"> A1. Pengguna menginjak pedal kiri pada <i>steering wheel</i>. <ol style="list-style-type: none"> 1. Mobil yang dikendarai pengguna pada aplikasi mengurangi kecepatan. 2. Kembali ke alur kejadian normal nomor 4. A2. Pengguna menggerakkan <i>steer</i> pada <i>steering wheel</i> ke kanan. <ol style="list-style-type: none"> 1. Mobil yang dikendarai pengguna pada aplikasi belok ke kanan. 2. Kembali ke alur kejadian normal nomor 4. A3. Pengguna menggerakkan <i>steer</i> pada <i>steering wheel</i> ke kiri. <ol style="list-style-type: none"> 1. Mobil yang dikendarai pengguna pada aplikasi belok ke kiri. 2. Kembali ke alur kejadian normal nomor 6. A4. Pengguna gagal menjalankan skenario aplikasi. <ol style="list-style-type: none"> 1. Aplikasi menampilkan pilihan restart dan home. 2. Kembali ke alur kejadian normal nomor 6.

3.2.3 Definisi Aktor

Aktor yang terdapat dalam aplikasi simulasi menyalip terlihat pada Tabel 3.5

Tabel 3.5 Deskripsi pengguna

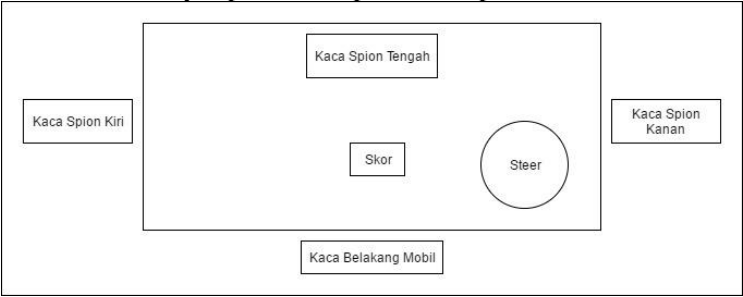
No	Nama	Deskripsi
1	Pengguna	Aktor yang bertugas untuk menggunakan dan menjalankan semua fungsionalitas pada aplikasi ini

3.2.4 Perancangan Model

Perancangan model pada aplikasi simulasi menyalip mobil ini diambil dari berbagai sumber seperti *Asset Store* Unity dan *website* penyedia model 3D secara gratis. Detil rancangan model ditampilkan pada subbab berikutnya.

3.2.4.1 Perancangan Model Interior Mobil

Terdapat beberapa model yang dibutuhkan. Yaitu kaca spion kanan, kiri, tengah, dan *steer*. Rancangan model interior pada aplikasi simulasi menyalip mobil dapat dilihat pada Gambar 3.2.






Gambar 3.2 Perancangan Model 3D Mobil Tampak Dalam

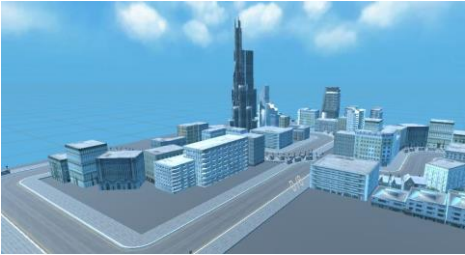

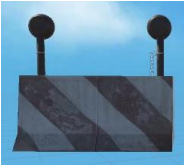
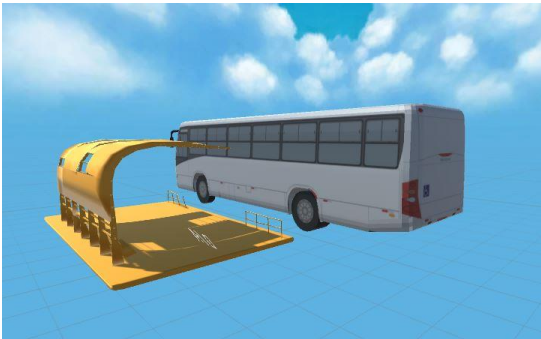
3.2.4.2 Daftar Asset

Dalam pembuatan aplikasi ini, banyak menggunakan *asset* seperti mobil, jalan dan juga bangunan – bangunan untuk menunjang

hasil yang bagus dan terlihat seperti nyata. Namun *asset* yang digunakan bukanlah hasil buatan sendiri namun merupakan asset gratis yang legal dan boleh digunakan. Beberapa asset yang digunakan disebutkan dalam Tabel 3.6.

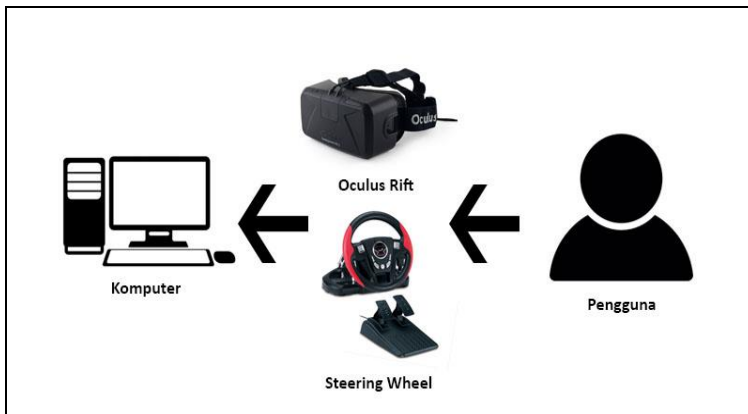
Tabel 3.6 Daftar Asset

No	Gambar	Sumber
1		https://free3d.com/3d-model/scion-frs-2013-high-poly-36878.html
2		https://free3d.com/3d-model/bmw-x5-1542.html
3		https://free3d.com/3d-model/street-system-v10-48448.html

No	Gambar	Sumber
4		https://free3d.com/3d-model/the-city-39441.html
5		https://free3d.com/3d-model/city-islands-63174.html
6		https://free3d.com/3d-model/road-barrier-34045.html
7		https://free3d.com/3d-model/marcopolo-senior-midi-19413.html

3.2.5 Arsitektur Umum Aplikasi

Arsitektur umum pada aplikasi simulasi menyalip mobil ini menggunakan beberapa perangkat tambahan. Antara lain adalah *Steering Wheel* dan *Oculus Rift*. Implementasi aplikasi ini dibuat menggunakan Unity. Untuk membuat dan mengedit objek 3D yang ada pada aplikasi digunakan *tools* Blender. Arsitektur secara umum aplikasi dapat dilihat pada Gambar 3.3.



Gambar 3.3 Perancangan Arsitektur Aplikasi

3.2.6 Perancangan Antarmuka Aplikasi

Rancangan antarmuka aplikasi diperlukan untuk memberikan gambaran umum kepada pengguna bagaimana sistem yang ada dalam aplikasi ini berinteraksi dengan pengguna. Selain itu, rancangan ini juga memberikan gambaran bagi pengguna apakah tampilan yang sudah disediakan oleh aplikasi mudah untuk dipahami dan digunakan, sehingga akan muncul kesan *user experience* yang baik dan mudah. Antarmuka yang akan dibuat diharapkan bisa memberikan pengalaman seperti di atas. Pada bagian ini dijelaskan mengenai perancangan antarmuka pada aplikasi simulasi menyalip mobil.

3.2.6.1 Antarmuka Menu Utama

Halaman antarmuka menu utama pada aplikasi ini muncul pada saat pertama kali pengguna membuka aplikasi ini. Pada antarmuka menu utama terdapat pilihan ujian yang direpresentasikan oleh beberapa tombol. Terdapat 4 tombol, 3 tombol merepresentasikan simulasi dengan masing-masing level dan satu tombol exit untuk keluar dari aplikasi. Rancangan antarmuka menu utama ditampilkan pada Gambar 3.4.

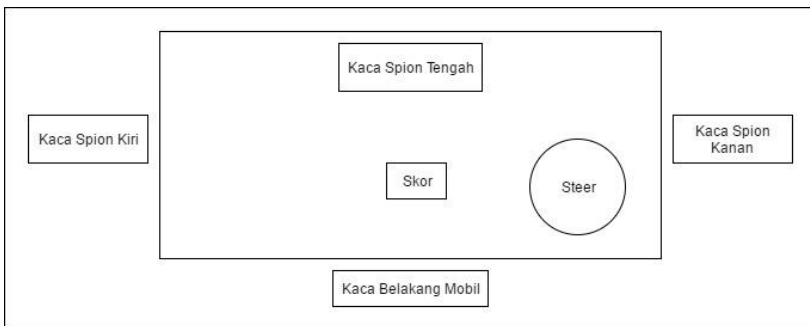


Gambar 3.4 Rancangan Antarmuka Menu Utama

3.2.6.2 Antarmuka Simulasi Menyalip Mobil

Halaman antarmuka simulasi menyalip mobil ini muncul ketika pengguna telah memilih salah satu dari beberapa pilihan level yang ada pada antarmuka menu utama. Ketika berada di antarmuka ini pengguna akan berada di dalam mobil pada tempat duduk pengemudi disebelah kanan. Terdapat beberapa tampilan pembantu yang ada pada antarmuka ini. Terdapat kaca spion tengah dan target menyalip mobil pada bagian atas. Kaca spion tengah berfungsi untuk menampilkan bagian belakang mobil, sementara skor merupakan target menyalip mobil yang harus dicapai agar dapat menyelesaikan

skenario Simulasi menyalip mobil ini. Terdapat kaca spion kanan pada bagian kanan yang berfungsi untuk memberi tampilan bagian kanan belakang pada mobil. Terdapat kaca spion kiri pada bagian kiri yang berfungsi untuk memberi tampilan bagian kiri belakang pada mobil. Yang terakhir terdapat *steering wheel* yang berada di depan pengguna yang berguna untuk menunjukkan state *steering wheel* sekarang. Tampilan rancangan antarmuka simulasi menyalip mobil ditampilkan pada Gambar 3.5.



Gambar 3.5 Rancangan Antarmuka Simulasi Menyalip Mobil

3.2.7 Perancangan Skenario Simulasi Menyalip Mobil

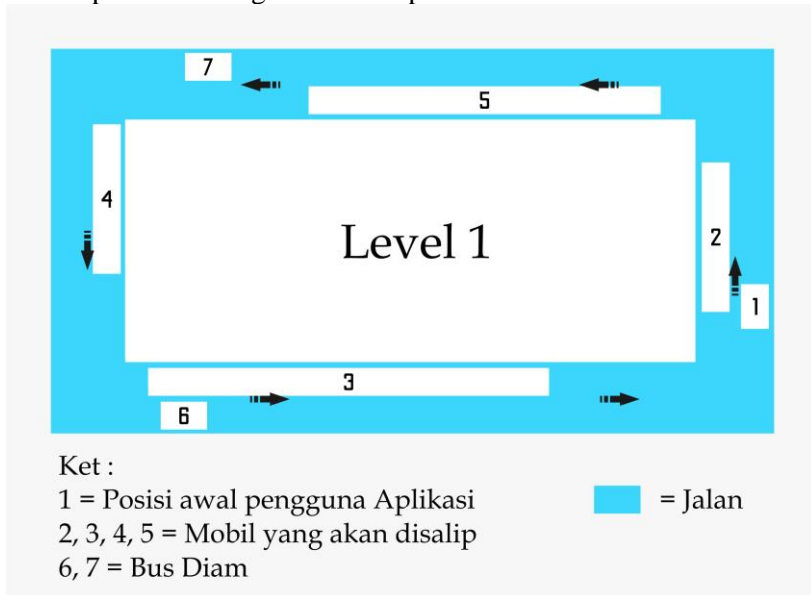
Pada subbab ini akan memberi gambaran mengenai skenario pada setiap ujian. Terdapat 3 level dalam skenario simulasi pada aplikasi ini. Terdapat level 1 yang mudah, level 2 yang sedang dan level 3 yang sulit. Pada skenario aplikasi ini pengguna memulai aplikasi dengan menyetir mobil. Pengguna akan langsung bisa merasakan fungsi dari steering wheel.

Pengguna diharuskan menyalip mobil sesuai dengan ketentuan disetiap levelnya. Apabila pengguna keluar dari jalanan maka scenario akan langsung berakhir (game over) dan pengguna memiliki dua pilihan yaitu mengulang level tersebut atau kembali ke menu utama. Selain keluar dari jalanan simulasi juga akan berakhir apabila mobil pengguna menabrak pembatas jalan yang berada di

setiap belokan dan menabrak mobil lawan baik yang searah maupun yang berlawanan arah dengan mobil pengguna. Untuk penjelasan lebih detailnya dapat dilihat pada subbab berikut ini.

3.2.7.1 Rancangan Skenario Simulasi Menyalip Mobil Level 1

Pada skenario ujian ini pengguna mengendarai mobil untuk bisa menyalip mobil lawan sebanyak 4 kali. Gambar 3.6 menampilkan rancangan skenario pada level 1.

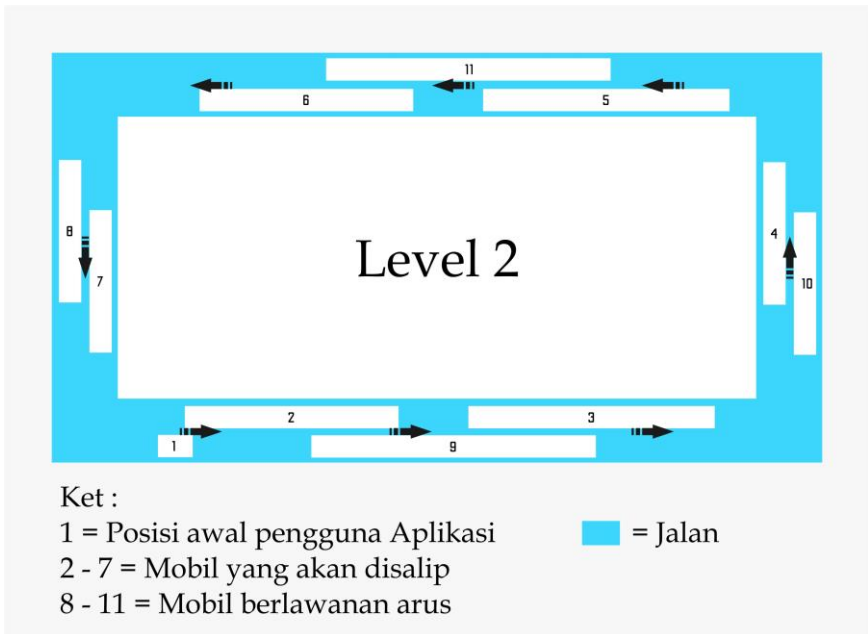


Gambar 3.6 Rancangan Skenario Level 1

Gambar di atas menunjukkan pada awal skenario, pengguna pertama kali berada di sisi luar jalan. Setelah itu pengguna akan menyalip mobil untuk bisa menyelesaikan level 1. Skenario akan langsung berhasil ketika mobil pengguna telah melewati mobil sebanyak 4 kali.

3.2.7.2 Rancangan Skenario Simulasi Menyalip Mobil Level 2

Pada skenario ujian ini pengguna mengendarai mobil untuk bisa menyalip mobil lawan sebanyak 4 kali. Gambar 3.7 menampilkan rancangan skenario pada level 2.

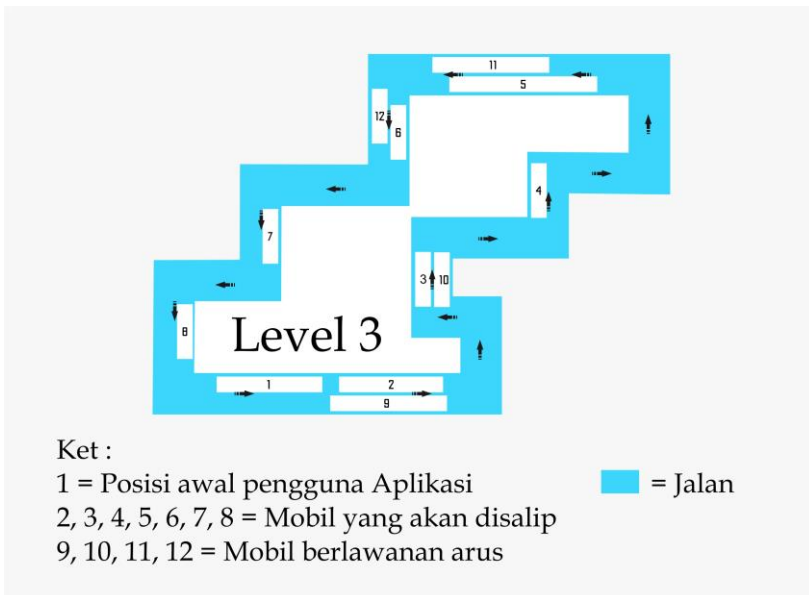


Gambar 3.7 Rancangan Skenario Level 2

Gambar diatas menjelaskan skenario pada level 2, sama seperti level 1 mobil user berada diluar jalanan. Setelah posisi awal, pengguna akan langsung mengendarai mobil dan untuk menyalip mobil supaya bisa menyelesaikan level 2. Apabila pengguna keluar dari jalanan maka scenario akan langsung berakhir. Skenario akan langsung berhasil ketika pengguna melewati mobil sebanyak 5 kali.

3.2.7.3 Rancangan Skenario Simulasi Menyalip Mobil Level 3

Skenario terakhir ini pengguna mengendarai mobil untuk bisa menyalip mobil lawan sebanyak 6 kali namun dengan jalan yang lebih sulit. Untuk memperjelas tampilan rancangan skenario pada level 3 ada pada Gambar 3.8.



Gambar 3.8 Rancangan Skenario Level 3

Pada gambar diatas dijelaskan level 3, sama dengan level 1 dan 2 mobil pengguna akan memulai star dari luar jalanan. Ketika pengguna masuk ke scenario ini, pengguna akan langsung mengendarai mobil yang digunakan untuk menyalip. Apabila pengguna keluar atau menabrak gedung disekeliling jalan, maka game akan langsung berakhir. Skenario akan berhasil jika pengguna melewati mobil sebanyak 6 kali.

BAB IV IMPLEMENTASI

Pada bab ini akan dibahas mengenai implementasi dari perancangan perangkat lunak. Cakupan implementasi dari perancangan perangkat lunak tersebut meliputi proses penerapan dan pengimplementasian algoritma dan antarmuka.

4.1 Lingkungan Implementasi

Lingkungan implementasi Tugas Akhir dijelaskan pada Tabel 4.1.

Tabel 4.1 Lingkungan Implementasi Perangkat Keras & Lunak

Perangkat Keras	Prosesor : - Intel® Core™ i3-2120 Processor CPU@3.30GHz Memori : - 4 GB Kartu Grafis : - Nvidia GeForce GTX 960
Perangkat Lunak	Sistem Operasi : - Microsoft Windows 10 Pro 64-bit Perangkat Pengembang : - Unity 5.3.4f1 - Visual Studio Ultimate 2012

4.2 Implementasi Antarmuka

Subbab ini membahas mengenai implementasi antarmuka pengguna yang meliputi implementasi antarmuka halaman menu utama, dan antarmuka halaman simulasi ujian sim.

4.2.1 Implementasi Antarmuka Halaman Menu Utama

Tampilan pertama yang muncul pada aplikasi ini adalah antarmuka halaman menu utama dan di implementasikan di scene MainMenu.unity. Pada antarmuka halaman menu utama terdapat logo Simulasi Menyalip Mobil. Lalu terdapat 4 tombol tepat berada di bawah logo. Tombol satu sampai ke tiga adalah tombol yang

merepresentasikan setiap level yang ada pada aplikasi ini. Sedangkan tombol ke 6 adalah tombol *exit* yang berfungsi untuk keluar dalam aplikasi ini. Tampilan antarmuka halaman menu utama dapat dilihat pada Gambar 4.1.



Gambar 4.1 Antarmuka Halaman Menu Utama

Pada setiap tombol yang ada pada antarmuka halaman menu utama terdapat kode yang membuat aplikasi berpindah *scene* sesuai dengan tombol yang dipilih. Tombol ‘Level 1’, ‘Level 2’ dan ‘Level 3’ akan membuat *scene* berpindah ke simulasi Level yang dipilih. Sedangkan tombol ‘Exit’ bertugas untuk keluar aplikasi. Implementasi halaman menu utama dapat dijelaskan pada Kode Sumber 4.1 yang berada pada kelas `MainMenuButton`.

```

1. using UnityEngine;
2. using UnityEngine.UI;
3. using System.Collections;
4.
5. public class MainMenuButton: MonoBehaviour {
6.
7.     public Button lvl1, lvl2, lvl3, exit;
8.

```

```

9.      void Start()
10.     {
11.         Button btn1 = lvl1.GetComponent<Button>();
12.         btn1.onClick.AddListener(Button1);
13.
14.         Button btn2 = lvl2.GetComponent<Button>();
15.         btn2.onClick.AddListener(Button2);
16.
17.         Button btn3 = lvl3.GetComponent<Button>();
18.         btn3.onClick.AddListener(Button3);
19.
20.         Button btnexit = exit.GetComponent<Button>();
21.         btnexit.onClick.AddListener(Exit);
22.     }
23.
24.     public void Button1() {
25.         Application.LoadLevel("Level1");
26.     }
27.
28.     public void Button2()
29.     {
30.         Application.LoadLevel("Level2");
31.     }
32.
33.     public void Button3()
34.     {
35.         Application.LoadLevel("Level3");
36.     }
37.
38.     public void Exit()
39.     {
40.         Application.Quit();
41.     }
42. }

```

Kode Sumber 4.1 Kode Sumber Antarmuka Menu Utama

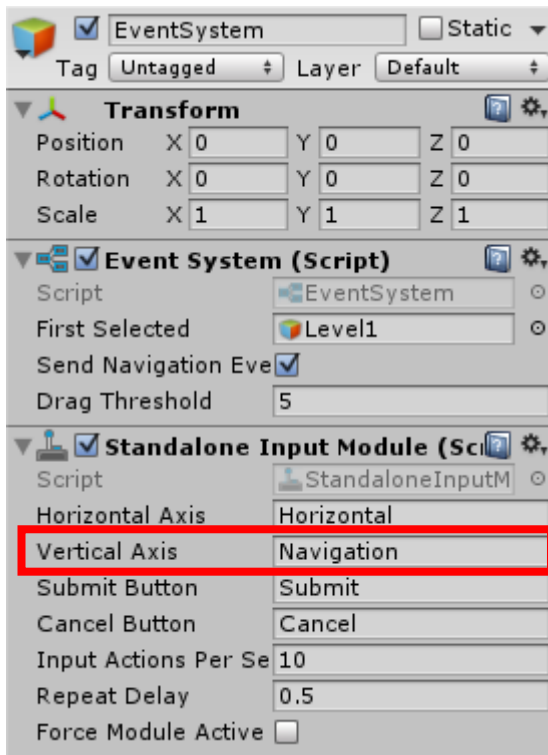
Pada antarmuka di halaman menu utama, tombol akan berpindah apabila *steering wheel* digerakkan ke kanan atau ke kiri. Prosesnya sama dengan menggerakkan mobil ke kanan dan ke kiri.

Oleh karena itu dibutuhkan input baru pada input manager [2] yang memiliki *type* berupa *Joystick Axis* karena inputan berupa *joystick* yaitu *steering wheel* dan *Axis* berupa *X Axis* karena belok ke kanan dan ke kiri merupakan *X axis*. Pada implementasi input baru ini diberi nama *Navigation*. Lalu *vertical axis* pada *standalone input module (Script)* yang berada pada *EventSystem* dirubah ke inputan yang baru dibuat yaitu *Navigation*. Sehingga perpindahan antar button bisa menggunakan *steering wheel*. Pembuatan *input* pada *input manager* akan dijelaskan pada Gambar 4.2. Lalu perubahan yang ada pada *EventSystem* akan dijelaskan di Gambar 4.3.

The image shows a Unity Input Manager window with a dropdown menu set to 'Navigation'. The configuration is as follows:

Property	Value
Name	Navigation
Descriptive Name	
Descriptive Negative Name	
Negative Button	left
Positive Button	right
Alt Negative Button	a
Alt Positive Button	d
Gravity	3
Dead	0.001
Sensitivity	3
Snap	<input checked="" type="checkbox"/>
Invert	<input checked="" type="checkbox"/>
Type	Joystick Axis
Axis	X axis
Joy Num	Get Motion from all Joysticks

Gambar 4.2 Pembuatan *Input Navigation*



Gambar 4.3 Perubahan Pada *EventSystem*

4.2.2 Implementasi Antarmuka Halaman Simulasi

Pada Antarmuka ini akan muncul pada setiap skenario simulasi ujian. Karena pada setiap antarmuka ujian memiliki tampilan yang sama, maka semuanya akan dijelaskan pada subbab ini. Tampilan antarmuka halaman simulasi ujian akan diperlihatkan pada Gambar 4.4.



Gambar 4.4 Antarmuka Halaman Simulasi

Pada antarmuka ini terdapat lima bagian yang penting pada antarmuka ini. *Steer* pada mobil, kaca spion tengah, kaca spion kiri, kaca spion kanan dan skor jumlah mobil yang dilewati. *Steer* pada mobil merepresentasikan gerakan *steering wheel* yang digunakan untuk belok kiri dan kanan. Nilai skor menyalip yang memberikan keterangan sudah berapa mobil yang disalip. Kaca spion kiri berfungsi sebagai kaca spion di sebelah kiri mobil. Kaca spion kanan berfungsi sebagai kaca spion di sebelah kanan mobil. Kaca spion tengah berfungsi sebagai kaca spion di sebelah tengah atas mobil. Semua bagian ini untuk membantu pengguna saat melakukan simulasi menyalip mobil. Dan semua bagian ini terdapat pada mobil seperti pada umumnya.

4.2.2.1 Implementasi *steer* pada mobil

Pada implementasi bagian ini menggunakan alat *steering wheel* dari Genius Speed Wheel 6 MT. Oleh karena itu, *steer* pada mobil dalam aplikasi harus mengikuti pergerakan dari alat *steering wheel*. Tampilan *steer* pada mobil dapat dilihat pada Gambar 4.5.



Gambar 4.5 Tampilan *steer* pada mobil

Maka dari itu untuk mengikuti pergerakan dari alat *steering wheel*, diperlukan kode sumber yang di taruh di *steer* pada mobil. Implementasi *steer* pada mobil dapat dijelaskan pada Kode Sumber 4.2 yang berada pada kelas CarControl.

```

1. void UpdateSteerAngle()
2. {
3.     float steer = Input.GetAxis("Horizontal") * -
    90;
4.     float posy = car.transform.eulerAngles.y;
5.     float posx = car.transform.eulerAngles.x;
6.     SteerWheel.transform.eulerAngles = new Vector3(po
    sx, posy, steer);
7. }

```

Kode Sumber 4.2 Kode Sumber *steer* pada mobil

4.2.2.2 Implementasi Skor Menyalip

Tempat skor ini bertujuan untuk memberitahukan kepada pengguna jumlah mobil yang telah di salip. Pada bagian kanan garis miring (/) di tempat skor berguna sebagai informasi bagi pengguna target jumlah mobil yang harus disalip. Sedangkan nilai dibagian sebelah kiri adalah nilai skor saat ini dan akan bertambah ketika mobil pengguna menyalip mobil lain. Apabila pengguna mencapai target nilai yang tertera, maka pengguna dikatakan berhasil menjalan simulasi menyalip ini. Tampilan skor pada mobil dapat dilihat pada Gambar 4.6.



Gambar 4.6 Tampilan Skor Menyalip

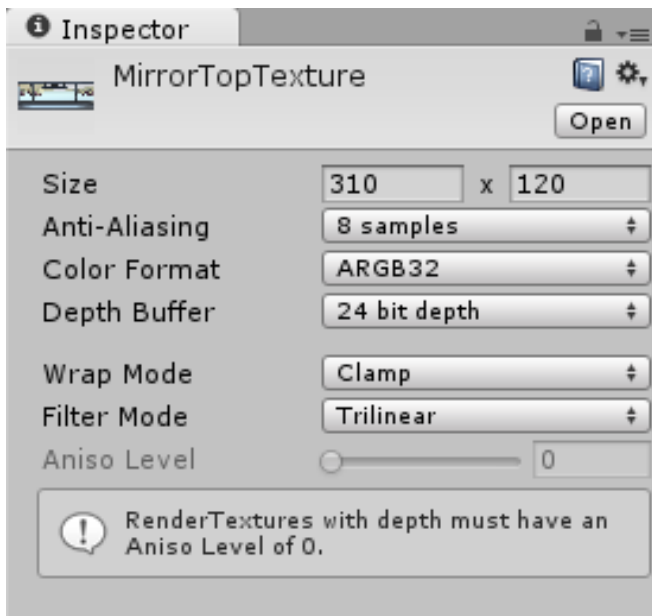
4.2.2.3 Implementasi Kaca Spion Tengah

Kaca spion tengah bertujuan untuk merepresentasikan kaca spion tengah atas yang berada pada mobil. Dengan adanya kaca spion tengah ini pengguna akan lebih mudah melihat bagian belakang mobil. Tampilan kaca spion tengah dapat dilihat pada Gambar 4.7

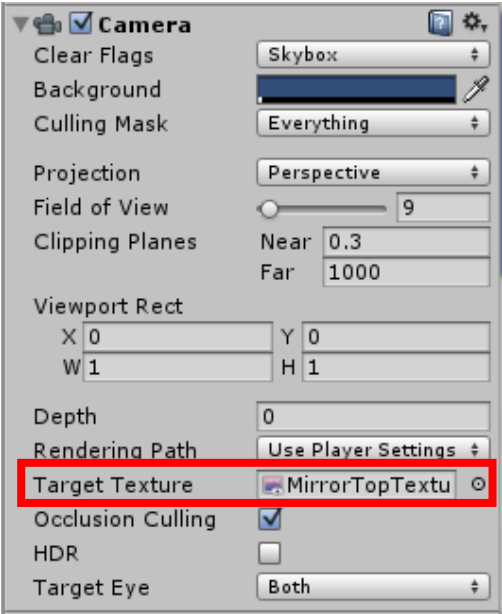


Gambar 4.7 Tampilan Kaca Spion Tengah

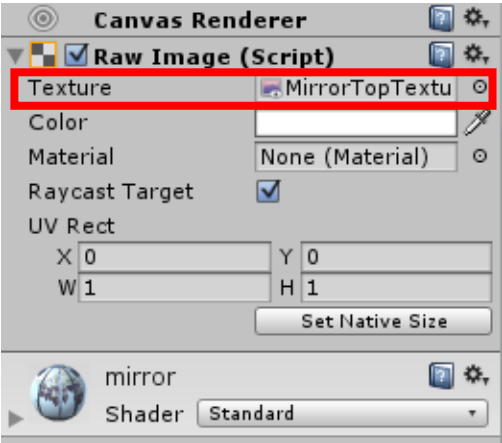
Kaca spion tengah ini di implementasikan dengan cara menambahkan kamera pada kaca spion yang sudah ada materialnya, apabila belum ada materialnya maka dapat dibuat dulu secara manual. Kamera yang telah ditambahkan tadi diletakan menghadap kebelakang sehingga kamera itu memperlihatkan seolah olah kaca yang menghadap kebelakang mobil. Setelah itu, tambahkan *raw image* pada *canvas UI* agar dapat menampilkan gambar yang kamera rekam. Sebelum itu buatlah *render texture* baru yang memiliki ukuran sama dengan *raw image*-nya. Pembuatan *texture render* baru dapat dilihat pada Gambar 4.8. Pada kamera harus diganti value *target texture* menjadi *texture render* yang tadi dibuat. Pembuatan kamera dapat dilihat pada Gambar 4.9. Lalu tambahkan *raw image* baru yang diletakkan sama dengan kamera. Value *texture* pada *raw image* akan diganti dengan *render texture* yang tadi telah dibuat. Pembuatan *raw image* dapat dilihat pada Gambar 4.10.



Gambar 4.8 Pembuatan *Texture Render* Kaca Spion Tengah



Gambar 4.9 Pembuatan Kamera Kaca Spion Tengah



Gambar 4.10 Pembuatan Raw Image Kaca Spion Tengah

4.2.2.4 Implementasi Kaca Spion Kanan

Kaca spion kanan bertujuan untuk merepresentasikan kaca spion kanan yang berada pada mobil. Dengan adanya kaca spion kanan ini pengguna akan lebih mudah melihat bagian belakang kanan mobil. Tampilan pada kaca spion kanan dapat dilihat pada Gambar 4.11.



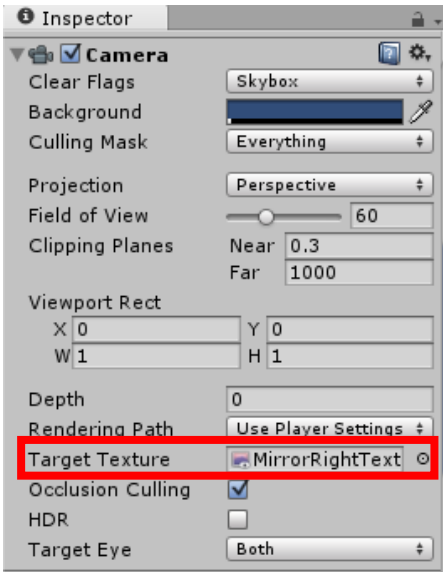
Gambar 4.11 Tampilan Kaca Spion Kanan

Implementasi Kaca spion kanan ini hampir sama dengan kaca spion tengah. Pertama tambahkan kamera disebelah kanan yang menghadap ke belakang sehingga kamera itu memperlihatkan seolah olah kaca yang menghadap kebelakang. lalu dengan adanya *raw image* pada *canvas UI* maka dapat menampilkan gambar yang terekam oleh kamera tersebut. Sebelum itu buatlah *render texture* baru yang memiliki ukuran sama dengan *raw image*-nya. Pembuatan *texture render* baru dapat dilihat pada Gambar 4.12. Lalu dibuatlah kamera baru yang diletakkan menghadap belakang dan diletakkan di tengah atas pada dalam mobil. Lalu ganti value *target texture* menjadi *texture render* yang tadi dibuat sebelumnya. Pembuatan kamera dapat dilihat pada Gambar 4.13. Lalu dibuatlah *raw image* baru yang diletakkan sama dengan kamera yang tadi dibuat. Value

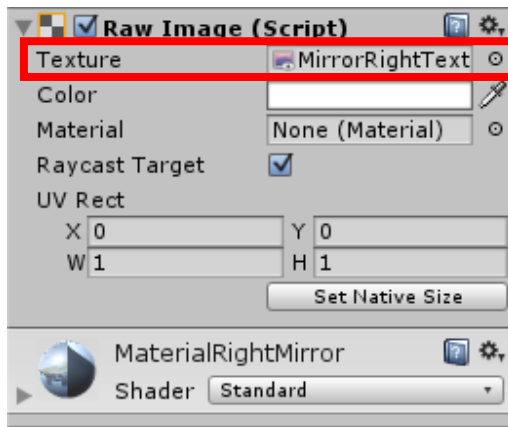
texture pada *raw image* akan diganti dengan *render texture* yang tadi telah dibuat. Pembuatan *raw image* dapat dilihat pada Gambar 4.14.



Gambar 4.12 Pembuatan *Texture Render Kaca Spion Kanan*



Gambar 4.13 Pembuatan Kamera Kaca Spion Kanan



Gambar 4.14 Pembuatan *Raw Image* Kaca Spion Kanan

4.2.2.5 Implementasi Kaca Spion Kiri

Kaca spion kiri bertujuan untuk merepresentasikan kaca spion kiri yang berada pada mobil. Dengan adanya kaca spion kiri ini pengguna akan lebih mudah melihat bagian belakang kiri mobil. Tampilan kaca spion kiri dapat dilihat pada Gambar 4.15.

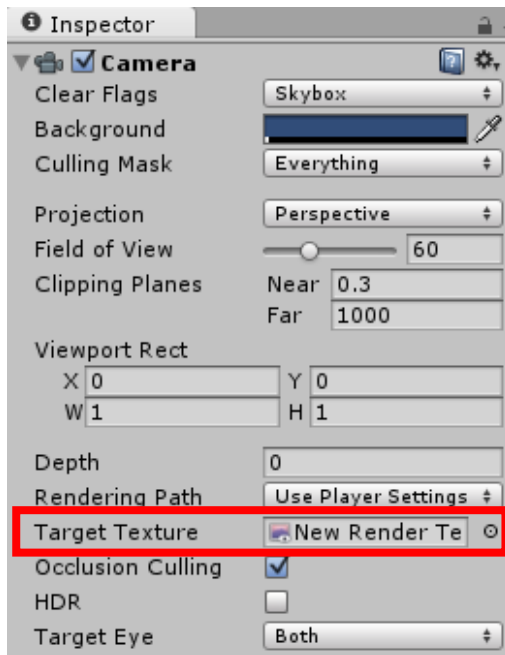


Gambar 4.15 Tampilan Kaca Spion Kiri

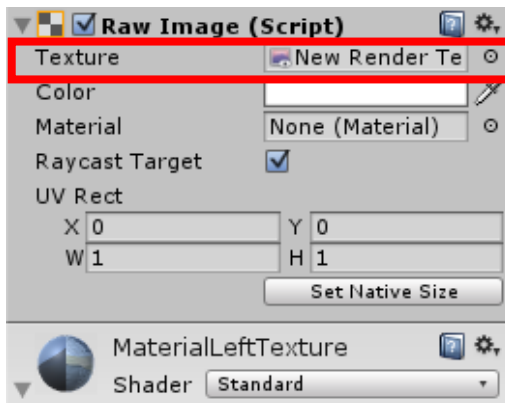
Kaca spion kiri ini di implementasikan dengan meletakkan kamera yang menghadap ke belakang sehingga kamera itu memperlihatkan seolah olah kaca yang menghadap kebelakang. lalu dengan *raw image* pada *canvas UI* dapat menampilkan gambar yang terekam oleh kamera tersebut. Sebelum buatlah *render texture* baru yang memiliki ukuran sama dengan *raw image*-nya. Pembuatan *texture render* baru dapat dilihat pada Gambar 4.16. Lalu dibuatlah kamera baru yang diletakkan menghadap belakang dan diletakkan di tengah atas pada dalam mobil. Lalu ganti value *target texture* menjadi *texture render* yang tadi dibuat sebelumnya. Pembuatan kamera dapat dilihat pada Gambar 4.17. Lalu dibuatlah *raw image* baru yang diletakkan sama dengan kamera yang tadi dibuat. Value *texture* pada *raw image* akan diganti dengan *render texture* yang tadi telah dibuat. Pembuatan *raw image* dapat dilihat pada Gambar 4.18.



Gambar 4.16 Pembuatan *Texture Render* Kaca Spion Kiri



Gambar 4.17 Pembuatan Kamera Kaca Spion Kiri



Gambar 4.18 Pembuatan Raw Image Kaca Spion Kiri

4.2.2.6 Implementasi Gear Maju Mundur

Untuk implementasi perubahan gear maju dan mundur ini menggunakan di alat *steering wheel*. Untuk bisa diimplementasikan tambahkan *input* baru yang dibuat di *input manager* yang terdapat pada File-Project Setting-input. Untuk penambahan *input* baru pada *input manager* dapat dilihat pada Gambar 4.23. Sebelum bisa berfungsi tambahkan pada script yang dijelaskan pada gambar Kode Sumber 4.3 yang berada pada kelas CarControl.



Gambar 4.19 Penambah *Input* Untuk Perubahan Gear

```

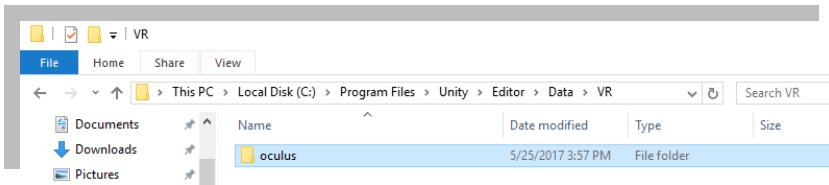
1. void UpdateGear ()
2. {
3.     bool maju = Input.GetButton("maju");
4.     bool mundur = Input.GetButton("mundur");
5.     if(maju)
6.     {
7.         Tran = true;
8.     }
9.     else if(mundur)
10.    {
11.        Tran = false;
12.    }
13. }

```

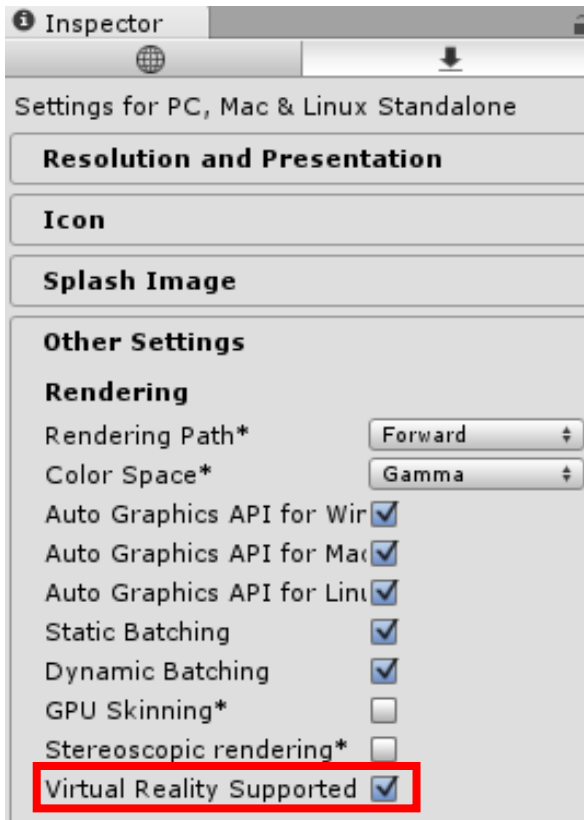
Kode Sumber 4.3 Kode Sumber Perubahan Teks Gear

4.3 Implementasi Integrasi Oculus Dengan Unity

Untuk pembuatan aplikasi ini menggunakan unity sebagai *platformnya* dan oculus rift sebagai *tools* yang digunakan untuk menjalankan aplikasi. Oleh karena itu, maka diperlukan integrasi agar oculus rift dapat digunakan saat membuat aplikasi ini. Implementasi integrasi oculus dengan unity pada saat membuat aplikasi ini adalah menggunakan OVRplugin for unity 1.3.2 yang dapat diunduh di <https://developer.oculus.com/downloads/>[3]. Hasil unduh dari berkas di atas adalah sebuah *package* unity. Untuk bisa membuka *package* tersebut harus *diimport* terlebih dahulu ke dalam *project* unity. Dalam *package* tersebut terdapat *folder* plugin yang akan dimasukkan ke direktori unity. Folder oculus tersebut lalu di *copy paste* ke direktori Unity/Editor/Data/VR. Tampilan direktori yang dipakai dapat dilihat pada Gambar 4.20. Setelah itu oculus siap digunakan pada unity. Terakhir sebelum mulai menggunakan oculusnya adalah mengizinkan unity untuk mengakses *virtual reality* pada *player setting*. Tampilan *player setting* dapat di lihat pada Gambar 4.21.



Gambar 4.20 Tampilan Direktori yang Digunakan



Gambar 4.21 Tampilan *Player Setting*

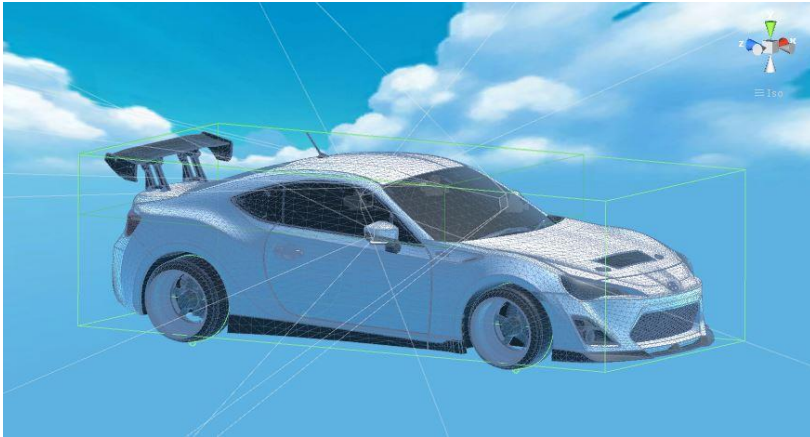
4.4 Implementasi Sistem Kerja Mobil

Sistem kerja mobil ini adalah salah satu hal yang paling penting. Bagaimana mobil dalam aplikasi ini dapat berjalan sesuai dengan mobil sesungguhnya. Objek mobil yang terdapat pada simulasi ini adalah mobil dari luar Indonesia, dimana steernya berada di sebelah kiri mobil. Oleh karena itu untuk bisa digunakan pada simulasi ini, mobil dilakukan pembalikan objek supaya sisi kiri objek berubah jadi sisi kanan dan sebaliknya, efek dari pembalikan ini nilai yang terdapat di mobil ini nantinya juga akan mengalami perbalikan.

Untuk bisa merasakan sentuhan dari benda lain, maka mobil ini memerlukan collider yang dibuat disekitar objek mobil. Terdapat 2 *collider* yang digunakan pada mobil pengguna, yang pertama adalah *box collider* yang ukurannya sesuai ukuran mobil dari kiri, kanan, atas dan bawah yang digunakan sebagai *trigger* untuk mengetahui apabila bertabrakan dengan benda yang lain dan untuk menambah skor apabila bertabrakan dengan *collider* skor. Untuk bagian kedua adalah *wheel collider* berfungsi sebagai panyangga roda serta penggerak mobil. Sebagaimana roda menggerakkan mobil saat berjalan, maka *wheel collider* juga berfungsi sama.

Wheel collider [4] memiliki fungsi yang dapat memberikan gaya baik bergerak maupun berhenti. Kedua gaya tersebut adalah *motorTorque* dan *brakeTorque*. *MotorTorque* adalah gaya yang digunakan untuk *wheelcollider* bisa bergerak dan sedangkan *brakeTorque* digunakan untuk berhenti. Setelah di coba dengan beberapa nilai, akhirnya diputuskan untuk menggunakan nilai -600 untuk *motorTorque* dan -60.000 untuk *brakeTorque*. Tampilan *Box* dan *wheel collider* dapat dilihat pada Gambar 4.22.

Tentu hal terpenting dalam mobil adalah bagaimana mobil itu dapat bergerak dan dikemudikan oleh pengguna sesuai dengan mobil yang sesungguhnya. Maka diperlukan kode agar mobil dapat bergerak sesuai dengan mobil sesungguhnya. Implementasi sistem kerja mobil dapat dijelaskan pada Kode Sumber 4.4 yang berada pada kelas *CarControl*.



Gambar 4.22 Tampilan Box dan Wheel Collider

```

1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4. using UnityEngine.UI;
5.
6. public class CarControl1 : MonoBehaviour {
7.
8.     float MotorF = -500;
9.     float BrakeF = -10000;
10.
11.     public int poin = 0;
12.     public TextMesh Check;
13.
14.     public WheelCollider[] wheelColliders = new WheelCollider[4];
15.     public Transform [] tiremeshes = new Transform[4];
16.     public Transform centerOfMassBody;
17.     public GameObject SteerWheel;
18.     public Rigidbody car;
19.     bool Tran = true;
20.     public float speed = 0;
21.     public float accelerate;

```

```

22.
23. void Start () {
24.     car.centerOfMass = centerOfMassBody.transform.localPosition;
25. }
26.
27. void Update () {
28.     Check.text = "" + poin;
29.     UpdateSteerAngle();
30.     UpdateWheelRotation();
31.     UpdateGear();
32.     if (poin == 4) {
33.         Application.LoadLevel("Next1");
34.     }
35. }
36.
37. void FixedUpdate()
38. {
39.     accelerate = Input.GetAxis("Vertical") + Input.GetAxis("Vertical2");
40.     UpdateWheelAngle();
41.     print(car.velocity);
42.     if(Tran == true)
43.     {
44.         wheelColliders[0].brakeTorque = 0;
45.         wheelColliders[1].brakeTorque = 0;
46.         wheelColliders[2].brakeTorque = 0;
47.         wheelColliders[3].brakeTorque = 0;
48.         if(accelerate > 0)
49.         {
50.             print("Maju");
51.             ForwardAccelerate();
52.         }
53.         else if (accelerate == 0)
54.         {
55.             print("Netral");
56.             Netral();
57.         }
58.         else if (accelerate < 0)
59.         {
60.
61.             print("Ngerem");
62.             Brake();

```

```

63.         }
64.     }
65.
66.     else if (Tran == false)
67.     {
68.         BackwardAccelerate();
69.         if(accelerate >= 0)
70.         {
71.             wheelColliders[0].brakeTorque = 0;
72.             wheelColliders[1].brakeTorque = 0;
73.             wheelColliders[2].brakeTorque = 0;
74.             wheelColliders[3].brakeTorque = 0;
75.         }
76.         else if (accelerate < 0)
77.         {
78.             Brake();
79.         }
80.     }
81.
82. void OnTriggerEnter(Collider other) {
83.     if (other.gameObject.tag.Equals("Score")) {
84.         poin = poin + 1;
85.     }
86.
87.     if (other.gameObject.tag.Equals("GameOver")) {
88.         Application.LoadLevel("Over1");
89.     }
90. }
91.
92. void UpdateWheelRotation(){
93.     for(int i = 0; i < 4; i++)
94.     {
95.         Quaternion quat;
96.         Vector3 pos;
97.         wheelColliders[i].GetWorldPose(out pos, out quat);
98.         tiremeshes[i].position = pos;
99.         tiremeshes[i].rotation = quat;
100.     }
101. }
102.

```



```

103.     void UpdateGear()
104.     {
105.         bool maju = Input.GetButton("maju");
106.         bool mundur = Input.GetButton("mundur");
107.
108.         if(maju)
109.         {
110.             Tran = true;
111.         }
112.
113.         else if(mundur)
114.         {
115.             Tran = false;
116.         }
117.
118.     void UpdateSteerAngle(){
119.         float steer = Input.GetAxis("Horizontal")
120.         * 90;
121.         float posy = car.transform.eulerAngles.y;
122.         float posx = car.transform.eulerAngles.x;
123.
124.         SteerWheel.transform.eulerAngles = new Vector3(
125.         posx, posy, steer);
126.     }
127.     void UpdateWheelAngle()
128.     {
129.         float steer = Input.GetAxis("Horizontal")
130.         ;
131.         float finalAngle = steer * 50f;
132.         wheelColliders[0].steerAngle = finalAngle
133.         ;
134.         wheelColliders[1].steerAngle = finalAngle
135.         ;
136.     }
137.
138.     void Netral()
139.     {
140.         for (int i = 0; i < 4; i++)
141.         {
142.             wheelColliders[i].brakeTorque = -
143.             MotorF;

```

```

137.         }
138.     }
139.
140.     void ForwardAccelerate(){
141.         for (int i = 0; i < 4; i++)
142.         {
143.             wheelColliders[i].motorTorque = accel
erate * MotorF;
144.         }
145.     }
146.
147.     void Brake()
148.     {
149.         for (int i = 0; i < 4; i++)
150.         {
151.             wheelColliders[i].brakeTorque = (acce
lerate) * BrakeF;
152.         }
153.     }
154.
155.     void BackwardAccelerate()
156.     {
157.         for (int i = 0; i < 4; i++)
158.         {
159.             wheelColliders[i].motorTorque = -
accelerate * MotorF;
160.         }
161.     }
162. }

```

Kode Sumber 4.4 Kode Sumber Sistem Kerja Mobil

4.5 Implementasi Skenario Simulasi Menyalip Mobil

Dengan adanya scenario simulasi menyalip mobil ini, diharapkan pengguna dapat mengetahui apasaja yang dibutuhkan ketika akan menyalip mobil dan juga untuk membiasakan menyalip mobil di kehidupan sesungguhnya. Dalam implementasi ini ada 3 macam tingkatan level untuk dilalui pengguna, dari level mudah,

sedang dan sulit yang dituliskan dalam bentuk angka (level 1, level 2 dan level 3).

4.5.1 Implementasi Skenario Level 1

Implementasi pada skenario ujian level 1 ini sama dengan perancangan pada Gambar 3.6. Terdapat 4 mobil yang bisa didahului dan dua mobil bus berlawanan arus yang diam di halte. Pada awal simulasi pengguna akan langsung berada dalam mobil dengan skor 0 dari total 4. Untuk bisa menyelesaikan simulasi level 1 ini, pengguna harus bisa menyalip mobil dengan aman sebanyak 4 kali. Setiap mobil yang akan didahului memiliki *collider* sendiri yang apabila bersentuhan dengan *collider* mobil pengguna akan langsung membuat simulasi berakhir.

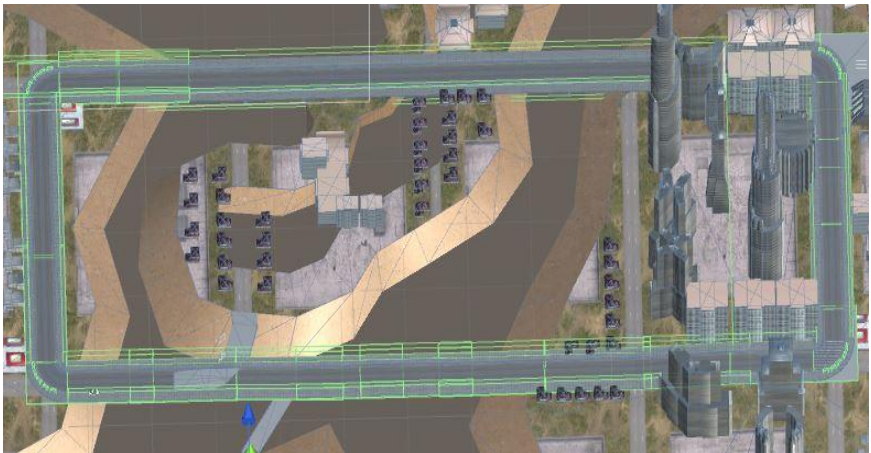
Untuk bagian sisi – sisi kiri dan kanan jalan juga terdapat *collider* yang digunakan sebagai pembatas untuk mobil pengguna tidak keluar dari jalanan. Selain dinding yang digunakan untuk membatasi jalan, juga terdapat *road barrier* yang diletakkan di belokan, tujuannya untuk membari tahu kepada pengguna bahwa ada belokan. Apabila penggukan menabrak *road barrier* ini maka akan langsung gagal dalam simulasi ini. Tampilan implementasi skenario level 1 dapat dilihat pada Gambar 4.23.



Gambar 4.23 Implementasi Skenario Level 1

4.5.2 Implementasi Skenario Level 2

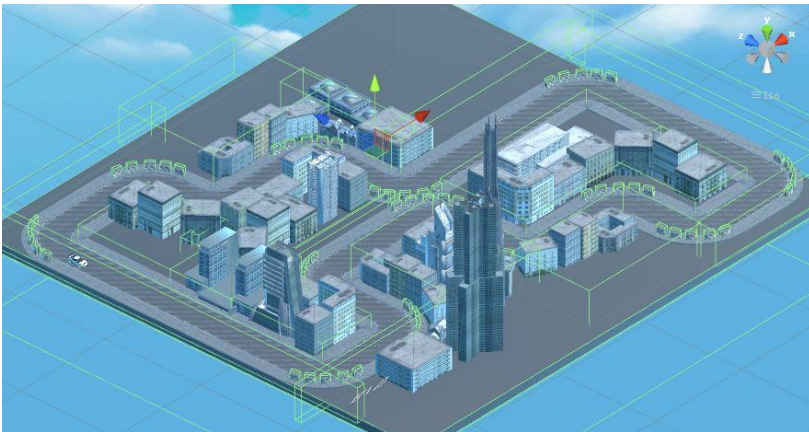
Implementasi skenario level 2 sama dengan perancangannya yang dapat dilihat pada Gambar 3.7. Terdapat 6 mobil yang bisa didahului dan 4 mobil yang berlawanan arus. Pada awal simulasi pengguna akan langsung berada dalam mobil dengan skor 0 dari total 4. Untuk bisa menyelesaikan simulasi level 2 ini, pengguna harus bisa menyalip mobil dengan aman sebanyak 4 kali. Setiap mobil yang akan didahului memiliki *collider* sendiri yang apabila bersentuhan dengan *collider* mobil pengguna akan langsung membuat simulasi berakhir. Sama dengan *collider* pada mobil lawan, pada sisi – sisi kiri dan kanan jalan juga terdapat *collider* yang digunakan sebagai pembatas untuk mobil pengguna tidak keluar dari jalanan. Selain dinding yang digunakan untuk membatasi jalan, juga terdapat *road barrier* yang diletakkan di belokan, tujuannya untuk membari tahu kepada pengguna bahwa ada belokan. Apabila penggukan menabrak *road barrier* ini maka akan langsung gagal dalam simulasi ini. Tampilan implementasi skenario level 2 dapat dilihat pada Gambar 4.24.



Gambar 4.24 Implementasi Skenario level 2

4.5.3 Implementasi Skenario Level 3

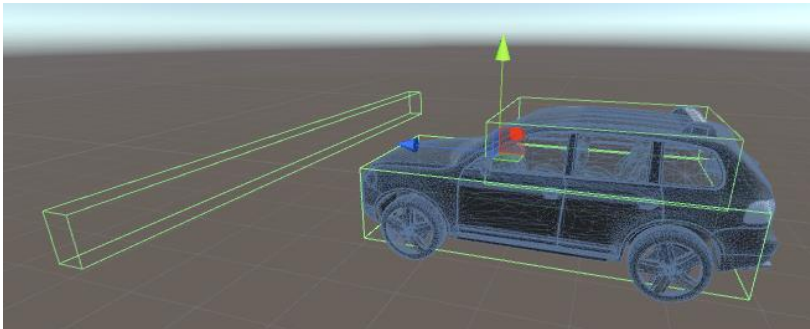
Implementasi pada skenario ujian level 3 ini sama dengan perancangan pada Gambar 3.8. Sama dengan dua level sebelumnya, pada level 3 juga terdapat mobil yang akan disalip sebanyak 7 mobil yang bisa didahului dan 4 mobil yang berlawanan arus. Pada awal simulasi level 3 ini pengguna akan langsung berada dalam mobil dengan skor 0 dari total 6. Untuk bisa menyelesaikan simulasi level 3 ini, pengguna harus bisa menyalip mobil sebanyak 6 kali. Setiap mobil yang akan didahului memiliki *collider* sendiri yang apabila bersentuhan dengan *collider* mobil pengguna akan langsung membuat simulasi berakhir. Sama dengan *collider* pada mobil lawan, pada sisi – sisi kiri dan kanan jalan juga terdapat *collider* yang digunakan sebagai pembatas untuk mobil pengguna tidak keluar dari jalanan. Selain dinding yang digunakan untuk membatasi jalan, juga terdapat *road barrier* yang diletakkan di belokan, tujuannya untuk membari tahu kepada pengguna bahwa ada belokan. Apabila penggunakan menabrak *road barrier* ini maka akan langsung gagal dalam simulasi ini. Tampilan implementasi skenario ujian maju mundur dapat dilihat pada Gambar 4.25.



Gambar 4.25 Implementasi Skenario Level 3

4.5.4 Implementasi Mobil yang akan Disalip

Sebelum membuat mobil bisa bergerak sesuai jalan pada setiap level, pertama buatlah mobil untuk bisa bergerak seperti mobil pengguna. Berbeda dengan mobil pengguna, mobil ini tidak membutuhkan *environment* yang lengkap, objek – objek yang dibutuhkan adalah badan mobil, dan rodanya. Pada mobil ini terdapat 2 *collider* yaitu *box collider* untuk badan mobil dan *wheel collider* untuk bergerak. Kode yang dipakai pada mobil lawan ini bisa dilihat di Kode Sumber 4.5. Terakhir supaya kemunculan mobil random maka dibutuhkan script baru yang berguna untuk mengacak kemunculannya berdasarkan nilai yang telah dimasukkan yang akan dijelaskan setelah ini. Untuk tampilan mobil yang disalip dapat dilihat pada gambar 4.26.



Gambar 4.26 Implementasi Mobil yang akan Disalip

```

1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4.
5. public class CarEngine : MonoBehaviour {
6.
7.     public Transform path;
8.     public float maxSteerAngle = 5000f;
9.     public WheelCollider wheelFL;
10.    public WheelCollider wheelFR;

```

```

11.     public float maxMotorTorque = 5000f;
12.     public float maxbrakeTorque = 3000f;
13.     public float currentSpeed;
14.     public float maxSpeed = 9000f;
15.     private float maksimal = 11000f;
16.     private List<Transform> nodes;
17.     private int currentNode = 0;
18.
19.     private void Start () {
20.         Transform[] pathTransforms = path.GetComponent<
    tsInChildren<Transform>());
21.         nodes = new List<Transform>();
22.
23.         for (int i = 0; i < pathTransforms.Length; i++)
24.         {
25.             if (pathTransforms[i] != path.transform){
26.                 nodes.Add(pathTransforms[i]);
27.             }
28.         }
29.
30.     private void FixedUpdate() {
31.         ApplySteer();
32.         Drive();
33.         CheckWaypointDistance();
34.     }
35.
36.     private void ApplySteer(){
37.         Vector3 relativeVector = transform.InverseTrans
    formPoint(nodes[currentNode].position);
38.         float newSteer = (relativeVector.x / relativeVe
    ctor.magnitude) * maxSteerAngle;
39.         wheelFL.steerAngle = newSteer;
40.         wheelFR.steerAngle = newSteer;
41.     }
42.
43.     private void Drive() {
44.         currentSpeed = 2 * Mathf.PI * wheelFL.radius *
    wheelFL.rpm * 60 / 1000;
45.         currentSpeed = 2 * Mathf.PI * wheelFR.radius *
    wheelFR.rpm * 60 / 1000;
46.

```

```

47.         if (currentSpeed < maxSpeed) {
48.             wheelFL.motorTorque = maxMotorTorque;
49.             wheelFR.motorTorque = maxMotorTorque;
50.         }
51.         else if (currentSpeed == maksimal){
52.             wheelFL.brakeTorque = maxbrakeTorque;
53.             wheelFR.brakeTorque = maxbrakeTorque;
54.         }
55.         else
56.         {
57.             wheelFL.motorTorque = 0;
58.             wheelFR.motorTorque = 0;
59.         }
60.     }
61.
62. private void CheckWaypointDistance() {
63.     if(Vector3.Distance(transform.position, nodes[c
        urrentNode].position) < 10f) {
64.         if(currentNode == nodes.Count - 1) {
65.             currentNode = 0;
66.         } else {
67.             currentNode++;
68.         }
69.     }
70. }
71. }

```

Kode Sumber 4.5 Kode Sumber Mobil yang Disalip

4.5.5 Implementasi Random Posisi Mobil

Pada implementasi random ini berguna untuk memunculkan mobil musuh secara acak. Random ini sendiri berguna supaya keadaan mobil pada saat jalanya aplikasi bisa seperti keadaan asli yaitu mobilnya bisa muncul dimanapun di jalanan. Untuk implementasinya dibutuhkan node – node yang telah tersambung bertujuan untuk sebagai jalur yang akan dilewati mobil nantinya yang disebut path. Tampilan implementasi node – node dapat dilihat pada gambar 4.27. Kode yang dipakai pada path ini dapat dilihat

pada Kode Sumber 4.6. Setelah terbentuk path, selanjutnya akan dirandom posisi dari mobil dengan random nilai x atau y atau z sesuai kebutuhan. Kode yang dipakai pada path ini dapat dilihat pada Kode Sumber 4.7.



Gambar 4.27 Implementasi Node – node rute mobil

```

1. using UnityEngine;
2. using System.Collections;
3. using System.Collections.Generic;
4. using System;
5.
6. public class Path : MonoBehaviour{
7.
8.     public Color lineColor;
9.
10.    private List<Transform> nodes = new List<Transform>();
11.
12.    void OnDrawGizmosSelected(){
13.        Gizmos.color = lineColor;
14.
15.        Transform[] pathTransforms = GetComponentsInChildren<Transform>();
16.        nodes = new List<Transform>();

```

```

17.
18.         for (int i = 0; i < pathTransforms.Length; i+
+)
19.         {
20.             if (pathTransforms[i] != transform)
21.             {
22.                 nodes.Add(pathTransforms[i]);
23.             }
24.         }
25.
26.         for (int i = 0; i < nodes.Count; i++)
27.         {
28.             Vector3 currentNode = nodes[i].position;
29.
30.             Vector3 previousNode = Vector3.zero;
31.
32.             if (i > 0)
33.             {
34.                 previousNode = nodes[i -
1].position;
35.             }
36.             else if (i == 0 && nodes.Count > 1)
37.             {
38.                 previousNode = nodes[nodes.Count -
1].position;
39.             }
40.             Gizmos.DrawLine(previousNode, currentNode
);
41.             Gizmos.DrawSphere(currentNode, 1f);
42.         }
43.     }
44.
45.     internal static string GetDirectoryName(string pa
thName)
46.     {
47.         throw new NotImplementedException();
48.     }
49.
50.     internal static string GetFileName(string pathNam
e)
51.     {
52.         throw new NotImplementedException();

```

```

53.     }
54.
55.     internal static string GetExtension(string fileName)
56.     {
57.         throw new NotImplementedException();
58.     }
59. }

```

Kode Sumber 4.6 Kode Sumber Mengabungkan Node – node

```

1. using UnityEngine;
2. using System.Collections;
3. using System.Collections.Generic;
4.
5. public class Random2 : MonoBehaviour
6. {
7.     public GameObject mobil;
8.
9.     void Start()
10.    {
11.        Vector3 updatePosisi = new Vector3(-
        462f, transform.position.y, UnityEngine.Random.Range(
        150f, 480f));
12.
13.        Quaternion originalRot = transform.rotation;
14.
15.        transform.rotation = originalRot * UnityEngine.Quaternion.AngleAxis(90, Vector3.up);
16.
17.        Instantiate(mobil, updatePosisi, transform.rotation);
18.    }
19. }

```

Kode Sumber 4.7 Kode Sumber Random Posisi Mobil

BAB V

PENGUJIAN DAN EVALUASI

Bab ini berisi bahasan mengenai uji coba dan evaluasi aplikasi simulasi menyalip mobil dengan menggunakan *Steering Wheel* dan Oculus Rift. Uji coba dilakukan berdasarkan skenario yang telah ditentukan. Uji coba dilakukan terhadap hasil implementasi perangkat lunak yang dijelaskan pada Bab IV.

5.1 Lingkungan Uji Coba

Proses uji coba dilakukan pada lingkungan yang telah ditentukan. Pada uji coba ini, lingkungan dibedakan menjadi lingkungan perangkat keras dan lingkungan perangkat lunak. Tabel 5.1 menjelaskan mengenai tiap-tiap lingkungan uji coba aplikasi.

Tabel 5.1 Lingkungan Pengujian Perangkat Keras dan Lunak

Perangkat Keras	Prosesor : - Intel® Core™ i3-2120 Processor CPU@3.30GHz Memori : - 4 GB Kartu Grafis : - Nvidia GeForce GTX 960
Perangkat Lunak	Sistem Operasi : - Microsoft Windows 10 Pro 64-bit Perangkat Pengembang : - Unity 5.3.4f1 - Visual Studio Ultimate 2012

5.2 Skenario Pengujian

Pada skenario pengujian dijelaskan tentang skenario pengujian yang dilakukan. Metode pengujian akan mengacu pada *blackbox testing*. Pengujian akan dibagi menjadi dua bagian, yaitu pengujian fungsionalitas dan pengujian aplikasi terhadap pengguna. Pengujian

ini dilakukan untuk mengetahui apakah fungsionalitas sistem telah berjalan sebagai mana mestinya.

5.2.1 Pengujian Fungsionalitas

Pengujian fungsionalitas dilakukan dengan menyiapkan beberapa skenario pengujian sebagai tolok ukur keberhasilan pengujian dan mengacu pada kasus penggunaan yang sudah dijelaskan pada 3.1.2.1. Metode pengujian yang digunakan pada pengujian fungsionalitas mengacu pada *blackbox testing*.

5.2.1.1 Memilih Tingkatan Simulasi Menyalip

Pengujian memilih Tingkat kesulitan dalam simulasi merupakan pengujian terhadap aplikasi untuk memilih level sesuai dengan keinginan pengguna. Pengguna dapat memilih satu dari 3 level yang disediakan. Skenario yang digunakan adalah pengguna disuruh untuk memilih salah satu dari 3 ujian yang berada pada antarmuka halaman menu utama. Apabila pengguna memilih level pertama, maka seharusnya *scene* harus berpindah ke level pertama dan begitu seterusnya. Gambar 5.1 Menjelaskan kondisi awal dalam memilih level di aplikasi menyalip mobil. Tabel 5.2 akan menjelaskan lebih lanjut perihal pengujian memilih level.



Gambar 5.1 Kondisi Awal dan Akhir Pengujian Memilih Level

Tabel 5.2 Pengujian Memilih Level Simulasi

Nomor	SP-UC-001
Referensi Kasus Penggunaan	UC-001
Nama	Memilih tingkat kesulitan Simulasi
Tujuan	Mengecek apakah aplikasi sudah dapat memilih level sesuai dengan apa yang dipilih oleh pengguna saat antarmuka halaman utama
Kondisi Awal	Pengguna membuka aplikasi
Skenario	Pengguna memilih salah satu dari tiga level yang tersedia
Keluaran yang Diharapkan	Aplikasi berpindah <i>scene</i> sesuai dengan level yang dipilih oleh pengguna
Hasil Pengujian	Berhasil

5.2.1.2 Menjalankan Simulasi Menyalip Mobil

Pengujian menjalankan simulasi menyalip mobil merupakan pengujian terhadap aplikasi untuk menjalankan skenario yang dipilih agar berjalan sebagaimana hasilnya. Terdapat 3 skenario level yang tersedia pada pengujian ini. Pertama pengguna menjalankan aplikasi secara gagal, kedua pengguna menjalankan aplikasi sebagaimana mestinya dengan tujuan menyelesaikan semua simulasi. Gambar 5.2 sampai 5.7 adalah contoh hasil pengujian memilih tingkat kesulitan simulasi menyalip mobil. Tabel 5.3 sampai tabel 5.8 akan menjelaskan lebih lanjut perihal pengujian memilih tingkat kesulitan simulasi.



Gambar 5.2 Kondisi Akhir Pengujian Secara Gagal

Tabel 5.3 Tabel Pengujian Menjalankan Simulasi Secara Gagal

Nomor	SP-UC-002
Referensi Kasus Penggunaan	UC-002
Nama	Menjalankan simulasi secara gagal
Tujuan	Mengecek apakah aplikasi sudah dapat memberikan hasil gagal apabila pengguna mengendarai mobil tidak sesuai dengan skenario. Contohnya adalah menabrak <i>road barrier</i> , menabrak mobil lain dan keluar dari jalan yang telah ditentukan.
Kondisi Awal	Pengguna memasuki <i>scene</i> skenario ujian
Skenario	Pengguna menabrak trotoar, menabrak mobil lain, jalan atau melewati jalan yang telah ditentukan.
Keluaran yang Diharapkan	Aplikasi berpindah <i>scene</i> yang menyatakan skenario ujian yang dilakukan telah gagal
Hasil Pengujian	Berhasil



Gambar 5.3 Kondisi Awal dan Akhir Pengujian Level 1

Tabel 5.4 Tabel Pengujian Menjalankan Level 1

Nomor	SP-UC-003
Referensi Kasus Penggunaan	UC-002
Nama	Menjalankan simulasi secara berhasil pada level 1
Tujuan	Aplikasi dengan tingkat kesulitan yang mudah sudah dapat memberikan hasil berhasil apabila pengguna menyalip mobil sebanyak target yang diminta.
Kondisi Awal	Pengguna memasuki <i>scene</i> skenario level 1.
Skenario	Pengguna menyalip mobil dengan target yang telah ditentukan. Pada pengujian ini adalah dengan tingkat kesulitannya yang mudah. Pengemudi akan mengendarai mobil sambil menyalip mobil lain tanpa menabrak mobil lain, pembatas jalan dan tidak keluar jalur.
Keluaran yang Diharapkan	Aplikasi berpindah <i>scene</i> yang menyatakan skenario simulasi yang dilakukan telah berhasil
Hasil Pengujian	Berhasil



Gambar 5.4 Kondisi Awal dan Akhir Pengujian Level 2

Tabel 5.5 Tabel Pengujian Menjalankan Level 2

Nomor	SP-UC-004
Referensi Kasus Penggunaan	UC-002
Nama	Menjalankan simulasi secara berhasil level 2
Tujuan	Aplikasi dengan tingkat kesulitan yang sedang sudah dapat memberikan hasil berhasil apabila pengguna menyalip mobil sebanyak target yang diminta.
Kondisi Awal	Pengguna memasuki <i>scene</i> skenario level 2.
Skenario	Pengguna menyalip mobil dengan target yang telah ditentukan. Pada pengujian ini adalah dengan tingkat kesulitannya yang sedang. Pengemudi akan mengendarai mobil sambil menyalip mobil lain tanpa menabrak mobil lain, pembatas jalan dan tidak keluar jalur.
Keluaran yang Diharapkan	Aplikasi berpindah <i>scene</i> yang menyatakan skenario simulasi yang dilakukan telah berhasil
Hasil Pengujian	Berhasil



Gambar 5.5 Kondisi Awal dan Akhir Pengujian Level 3

Tabel 5.6 Tabel Pengujian Menjalankan Level 3

Nomor	SP-UC-005
Referensi Kasus Penggunaan	UC-002
Nama	Menjalankan simulasi secara berhasil level 3
Tujuan	Aplikasi dengan tingkat kesulitan yang sulit sudah dapat memberikan hasil berhasil apabila pengguna menyalip mobil sebanyak target yang diminta.
Kondisi Awal	Pengguna memasuki <i>scene</i> skenario level 3
Skenario	Pengguna menyalip mobil dengan target yang telah ditentukan. Pada pengujian ini adalah dengan tingkat kesulitannya yang sulit. Pengemudi akan mengendarai mobil sambil menyalip mobil lain tanpa menabrak mobil lain, pembatas jalan dan tidak keluar jalu.
Keluaran yang Diharapkan	Aplikasi berpindah <i>scene</i> yang menyatakan skenario simulasi yang dilakukan telah berhasil
Hasil Pengujian	Berhasil

5.2.2 Pengujian Aplikasi Terhadap Pengguna

Aplikasi ini perlu dilakukan pengujian terhadap pengguna yang lainnya. Tujuannya adalah agar orang yang sudah atau belum bisa mengendarai mobil mengetahui cara dan suasana asli ketika menyalip mobil yang baik dan benar. Tujuan lainnya yaitu untuk mengetahui kesulitan dari tiap – tiap level yang ada dalam aplikasi saat digunakan oleh pengguna. Metode pengujian yang dilakukan pada pengujian aplikasi terhadap pengguna mengacu pada *blackbox testing*.

Pengujian dilakukan kepada sepuluh orang yang akan diminta untuk mengoperasikan aplikasi simulasi menyalip mobil. Setelah itu pengguna mengisi kuesioner yang telah disediakan. Kuesioner dapat dilihat pada Tabel 5.9. Masing-masing kolom akan memiliki bobot penilaian tersendiri dengan ketentuan sebagai berikut :

- Sangat Setuju = 5
- Setuju = 4
- Netral = 3
- Tidak Setuju = 2
- Sangat Tidak Setuju = 1

Tabel 5.7 Kuesioner Pengguna Simulasi Menyalip Mobil

No	Parameter Antarmuka	STS	TS	N	S	SS
1	Aplikasi memiliki tampilan dan desain yang menarik					
2	Aplikasi memiliki menu yang mudah digunakan					
3	Aplikasi Memiliki tata letak tombol yang sesuai					

	Parameter Performa dan Kenyamanan					
4	Aplikasi nyaman untuk digunakan					
	Parameter Materi Simulasi					
5	Aplikasi memberikan rasa mengendarai mobil seperti sesungguhnya					
6	Skenario pada aplikasi sama dengan yang sesungguhnya					

5.3 Evaluasi

Tahap evaluasi perlu di lakukan untuk mengolah hasil data yang telah dihasilkan. Lalu akan dihasilkan kesimpulan dan saran nantinya. Evaluasi dibagi menjadi dua bagian yaitu evaluasi pengujian fungsionalitas dan pengujian aplikasi terhadap pengguna.

5.3.1 Evaluasi Pengujian Fungsionalitas

Terdapat enam skenario pengujian fungsionalitas yang telah di lakukan pada tahap 5.2.1. Tabel 5.8 menunjukan hasil dari pengujian fungsionalitas. Semua skenario pengujian telah dicoba dan berhasil. Jadi dapat disimpulkan bahwa fungsionalitas dari aplikasi ini dapat berjalan sesuai dengan harapan.

Tabel 5.8 Tabel Hasil Evaluasi Pengujian Fungsionalitas

ID	Nama	Hasil
SP-UC-001	Memilih tingkat kesulitan Simulasi	Berhasil
SP-UC-002	Menjalankan simulasi secara gagal	Berhasil

ID	Nama	Hasil
SP-UC-003	Menjalankan simulasi secara berhasil pada level 1	Berhasil
SP-UC-004	Menjalankan simulasi secara berhasil pada level 2	Berhasil
SP-UC-005	Menjalankan simulasi secara berhasil pada level 3	Berhasil

5.3.2 Evaluasi Pengujian Aplikasi Terhadap Pengguna

Aplikasi ini telah diujikan kepada tujuh orang yang diambil secara acak. Namun diantara penguji tersebut harus ada minimal 2 orang yang belum bisa mengendarai mobil. Setelah mencoba aplikasi, pengguna diberikan kuesioner yang diisi berdasarkan pendapat mereka. Rentan umur pengguna adalah 20 sampai 22 tahun. Hasil dari Kuisoner dapat dilihat pada Tabel 5.9.

Tabel 5.9 Hasil Kuesioner Pengguna

No	Parameter Antarmuka	STS
1	Aplikasi memiliki tampilan dan desain yang menarik	4.2
2	Aplikasi memiliki menu yang mudah digunakan	4
3	Aplikasi Memiliki tata letak tombol yang sesuai	4
Parameter Performa dan Kenyamanan		
4	Aplikasi nyaman untuk digunakan	3.7

	Parameter Materi Simulasi	
5	Aplikasi memberikan rasa mengendarai mobil seperti sesungguhnya	3.7
No	Parameter Antarmuka	STS
6	Skenario pada aplikasi sama dengan yang sesungguhnya	3.5

Berdasarkan tabel diatas dibagi kedalam tiga bagian. Yaitu antarmuka, performa, kenyamanan dan materi simulasi. Pada penilaian antarmuka, aplikasi ini mendapatkan nilai rata – rata 4.1. Nilai ini sudah melebihi nilai 4 yaitu setuju. Hal ini menunjukkan bahwa pengguna setuju apabila aplikasi simulasi menyalip mobil ini memiliki antarmuka yang menarik, mudah digunakan, dan sesuai. Untuk nilai kenyamanan mendapatkan nilai 3.7. Nilai ini suda hampir mendekati 4 yaitu setuju. Hal ini menunjukkan bahwa pengguna mengatakan bahwa aplikasi ini cukup nyaman digunakan. Penilaian terakhir adalah nilai materi simulasi mendapatkan nilai 3.6. Nilai ini sudah lebih dari nilai 3 yaitu cukup. Hal ini menunjukkan aplikasi simulasi menyalip mobil ini memiliki materi simulasi yang cukup sesuai dengan sesungguhnya. Secara umum aplikasi ini mendapatkan nilai rata – rata 3.88 dari nilai maksimal 5.

BAB VI

KESIMPULAN DAN SARAN

Bab ini menjelaskan tentang kesimpulan yang didapatkan dari hasil pengujian yang telah dilakukan sebagai jawaban dari rumusan masalah yang telah dikemukakan sebelumnya. Setelah dari kesimpulan, diberikan juga saran sebagai pengembangan aplikasi kedepannya.

6.1. Kesimpulan

Dalam proses pengerjaan Tugas Akhir mulai dari tahap analisis, rancangan, implementasi, pengujian hingga evaluasi didapatkan kesimpulan sebagai berikut:

1. Ditinjau dari hasil uji coba, antarmuka aplikasi simulasi menyalip mendapatkan nilai 4.1 yang menunjukkan antarmuka yang cukup menarik dan mudah dipakai. Lalu mendapatkan nilai kenyamanan 3.7 yang berarti aplikasi cukup nyaman dipakai, walaupun kurang nyaman. Dan mendapatkan nilai materi 3.6 yang menunjukkan pengguna setuju bahwa aplikasi memiliki materi yang cukup sesuai dengan sesungguhnya
2. Dari tingkat kesulitan dalam simulasi menyalip, simulasi ini memiliki kesulitan yang bertingkat dari yang mudah, sedang dan susah. Kesulitan per level pada aplikasi ini berbagai macam dari jalan yang berbeda - beda sampai dengan jumlah mobil yang lebih banyak seiring dengan semakin sulitnya level.
3. Skor yang berada pada mobil membantu pengguna untuk mengetahui jumlah mobil yang telah disalip dan jumlah mobil yang harus disalip sesuai dengan ketentuan di setiap levelnya.

6.2. Saran

Berikut merupakan beberapa saran untuk pengembangan sistem di masa yang akan datang, berdasarkan pada hasil perancangan, implementasi dan uji coba yang telah dilakukan.


1. Steering wheel yang digunakan tidak mirip dengan steer pada mobil sesungguhnya. Steernya hanya bertupar 180^0 sedangkan pada mobil sesungguhnya 360^0 .
2. Untuk pengguna yang belum terbiasa menggunakan Oculus Rift, maka akan merasakan efek pusing.

DAFTAR PUSTAKA

- [1] X. Tang, Virtual Reality - Human Computer Interaction, InTech, Chapters published , 2012.
- [2] Unity. [Online]. Available: <http://docs.unity3d.com/Manual/class-InputManager.html>.
- [3] O. VR, "Developer Center - Downloads," [Online]. Available: <https://developer.oculus.com/downloads/>.
- [4] Unity, "Unity - Scripting API: Wheel Collider," [Online]. Available: <https://docs.unity3d.com/ScriptReference/WheelCollider.html>.
- [5] "Unity 3D - Game Engine | Herman Class," [Online]. Available: <http://www.hermantolle.com/class/docs/unity-3d-game-engine/>.

LAMPIRAN

79



ITS
Institut Teknologi
Sepuluh Nopember

KUISIONER TUGAS AKHIR – 5113100092 FAJAR ADE PUTRA

**RANCANG BANGUN SIMULASI MENYALIP MOBIL (OVERTAKING CAR) SESUAI DENGAN
PERATURAN PEMERINTAH 1993 MENGGUNAKAN STEERING WHEEL DAN OCULUS RIFT**

Identitas Responden

Nama Lengkap : Catya Angka M. Usia : 21

Pekerjaan : Mahasiswa Jenis Kelamin : ♂ / P

A. PENILAIAN TERHADAP APLIKASI

Silahkan tabel dibawah dengan menggunakan tanda (v)

SS = Sangat Setuju S = Setuju C = Cukup
TS = Tidak Setuju STS = Sangat Tidak Setuju

No	Parameter Antarmuka	STS	TS	C	S	SS
1	Aplikasi memiliki tampilan dan desain yang menarik					✓
2	Aplikasi memiliki menu yang mudah digunakan				✓	
3	Aplikasi Memiliki tata letak tombol yang sesuai				✓	
	Parameter Performa dan Kenyamanan					
4	Aplikasi nyaman untuk digunakan				✓	
	Parameter Materi Simulasi					
5	Aplikasi memberikan rasa mengendarai mobil seperti sesungguhnya				✓	
6	Skenario pada aplikasi sama dengan yang sesungguhnya				✓	

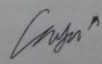
B. KRITIK DAN SARAN

Kontrol Untuk balok dikingkatkan


.....

.....

Surabaya, 4 Juni 2017


CATYA A.M.

Gambar 0. 1 Kuesioner Responden Pertama



ITS
Institut Teknologi
Sepuluh Nopember

KUISIONER TUGAS AKHIR – 5113100092 FAJAR ADE PUTRA

**RANCANG BANGUN SIMULASI MENYALIP MOBIL (OVERTAKING CAR) SESUAI DENGAN
PERATURAN PEMERINTAH 1993 MENGGUNAKAN STEERING WHEEL DAN OCULUS RIFT**

Identitas Responden

Nama Lengkap : Anwar Rosyidi Usia : 22

Pekerjaan : Mahasiswa Jenis Kelamin : P

A. PENILAIAN TERHADAP APLIKASI

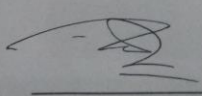
Isilah tabel dibawah dengan menggunakan tanda (v)
 SS = Sangat Setuju S = Setuju C = Cukup
 TS = Tidak Setuju STS = Sangat Tidak Setuju

No	Parameter Antarmuka	STS	TS	C	S	SS
1	Aplikasi memiliki tampilan dan desain yang menarik				<input checked="" type="checkbox"/>	
2	Aplikasi memiliki menu yang mudah digunakan				<input checked="" type="checkbox"/>	
3	Aplikasi Memiliki tata letak tombol yang sesuai				<input checked="" type="checkbox"/>	
	Parameter Performa dan Kenyamanan					
4	Aplikasi nyaman untuk digunakan				<input checked="" type="checkbox"/>	
	Parameter Materi Simulasi					
5	Aplikasi memberikan rasa mengendarai mobil seperti sesungguhnya			<input checked="" type="checkbox"/>		
6	Skenario pada aplikasi sama dengan yang sesungguhnya				<input checked="" type="checkbox"/>	


B. KRITIK DAN SARAN

mana tambah untuk pilihan di bagian samping
utama dasar

Surabaya, 2017


Anwar Rosyidi

Gambar 0. 2 Kesioner Responden Kedua



ITS
Institut Teknologi
Sepuluh Nopember

KUISIONER TUGAS AKHIR – 5113100092 FAJAR ADE PUTRA

**RANCANG BANGUN SIMULASI MENYALIP MOBIL (OVERTAKING CAR) SESUAI DENGAN
PERATURAN PEMERINTAH 1993 MENGGUNAKAN STEERING WHEEL DAN OCULUS RIFT**

Identitas Responden

Nama Lengkap : Muhammad Saddam Usia : 21

Pekerjaan : Mahasiswa Jenis Kelamin : ♂ P

A. PENILAIAN TERHADAP APLIKASI

Isilah tabel dibawah dengan menggunakan tanda (v)

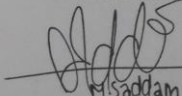
SS = Sangat Setuju S = Setuju C = Cukup
TS = Tidak Setuju STS = Sangat Tidak Setuju

No	Parameter Antarmuka	STS	TS	C	S	SS
1	Aplikasi memiliki tampilan dan desain yang menarik				✓	
2	Aplikasi memiliki menu yang mudah digunakan			✓		
3	Aplikasi Memiliki tata letak tombol yang sesuai				✓	
	Parameter Performa dan Kenyamanan					
4	Aplikasi nyaman untuk digunakan				✓	
	Parameter Materi Simulasi					
5	Aplikasi memberikan rasa mengendarai mobil seperti sesungguhnya			✓		
6	Skenario pada aplikasi sama dengan yang sesungguhnya				✓	


B. KRITIK DAN SARAN

- Asset dibuat lebih natural
- harap dapat lebih berinteraksi dgn user

Surabaya, 14 Juni 2017


M. Saddam

Gambar 0. 3 Kuesioner Responden Ketiga



ITS
 Institut Teknologi Sepuluh Nopember

KUISIONER TUGAS AKHIR – 5113100092 FAJAR ADE PUTRA

RANCANG BANGUN SIMULASI MENYALIP MOBIL (OVERTAKING CAR) SESUAI DENGAN PERATURAN PEMERINTAH 1993 MENGGUNAKAN STEERING WHEEL DAN OCULUS RIFT

Identitas Responden

Nama Lengkap : Fauziddarain Syam Usia : 24

Pekerjaan : Belasungkita Jenis Kelamin : ♂ P

A. PENILAIAN TERHADAP APLIKASI

Isilah tabel dibawah dengan menggunakan tanda (v)

SS = Sangat Setuju S = Setuju C = Cukup

TS = Tidak Setuju STS = Sangat Tidak Setuju

No	Parameter Antarmuka	STS	TS	C	S	SS
1	Aplikasi memiliki tampilan dan desain yang menarik					✓
2	Aplikasi memiliki menu yang mudah digunakan				✓	
3	Aplikasi Memiliki tata letak tombol yang sesuai				✓	
	Parameter Performa dan Kenyamanan					
4	Aplikasi nyaman untuk digunakan				✓	
	Parameter Materi Simulasi					
5	Aplikasi memberikan rasa mengendarai mobil seperti sesungguhnya				✓	
6	Skenario pada aplikasi sama dengan yang sesungguhnya				✓	

B. KRITIK DAN SARAN

Kalau ada guide jalannya lebih bagus.

Game over nya dibuat lebih susah ditambah perincasan

terlalu banyak monster

Surabaya, 04 Juni 2017

Fajar
Fauziddarain Syam

Gambar 0. 4 Kuesioner Responden Keempat



KUISIONER TUGAS AKHIR – 5113100092 FAJAR ADE PUTRA

RANCANG BANGUN SIMULASI MENYALIP MOBIL (OVERTAKING CAR) SESUAI DENGAN PERATURAN PEMERINTAH 1993 MENGGUNAKAN STEERING WHEEL DAN OCULUS RIFT

Identitas Responden

Nama Lengkap : Setya A Usia : 21
 Pekerjaan : Mahasiswa Jenis Kelamin : L / P

A. PENILAIAN TERHADAP APLIKASI

Isilah tabel dibawah dengan menggunakan tanda (v)

SS = Sangat Setuju

S = Setuju

C = Cukup

TS = Tidak Setuju

STS = Sangat Tidak Setuju


No	Parameter Antarmuka	STS	TS	C	S	SS
1	Aplikasi memiliki tampilan dan desain yang menarik				✓	
2	Aplikasi memiliki menu yang mudah digunakan				✓	
3	Aplikasi Memiliki tata letak tombol yang sesuai				✓	
Parameter Performa dan Kenyamanan						
4	Aplikasi nyaman untuk digunakan				✓	
Parameter Materi Simulasi						
5	Aplikasi memberikan rasa mengendarai mobil seperti sesungguhnya			✓		
6	Skenario pada aplikasi sama dengan yang sesungguhnya			✓		

B. KRITIK DAN SARAN

Perlu ditambahkan lagi bagian kontrol mobil sehingga
mirip dengan mobil ya sebenarnya

Surabaya, 2017

Gambar 0. 5 Kuesioner Responden Kelima



ITS
Institut
Teknologi
Sepuluh Nopember

KUISIONER TUGAS AKHIR – 5113100092 FAJAR ADE PUTRA

**RANCANG BANGUN SIMULASI MENYALIP MOBIL (OVERTAKING CAR) SESUAI DENGAN
PERATURAN PEMERINTAH 1993 MENGGUNAKAN STEERING WHEEL DAN OCULUS RIFT**

Identitas Responden

Nama Lengkap : A. Ade Usia : 22

Pekerjaan : Mahasiswa Jenis Kelamin : L / P

A. PENILAIAN TERHADAP APLIKASI

Isilah tabel dibawah dengan menggunakan tanda (v)

SS = Sangat Setuju S = Setuju C = Cukup
TS = Tidak Setuju STS = Sangat Tidak Setuju

No	Parameter Antarmuka	STS	TS	C	S	SS
1	Aplikasi memiliki tampilan dan desain yang menarik				✓	
2	Aplikasi memiliki menu yang mudah digunakan				✓	
3	Aplikasi Memiliki tata letak tombol yang sesuai				✓	
	Parameter Performa dan Kenyamanan					
4	Aplikasi nyaman untuk digunakan			✓		
	Parameter Materi Simulasi					
5	Aplikasi memberikan rasa mengendarai mobil seperti sesungguhnya				✓	
6	Skenario pada aplikasi sama dengan yang sesungguhnya			✓		

B. KRITIK DAN SARAN

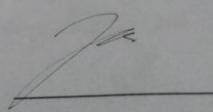
Belum ada fitur yang lengkap

.....

.....

.....

Surabaya, 5 Juni 2017



Gambar 0. 6 Kuesioner Responden Keenam



KUISIONER TUGAS AKHIR – 5113100092 FAJAR ADE PUTRA

RANCANG BANGUN SIMULASI MENYALIP MOBIL (OVERTAKING CAR) SESUAI DENGAN
PERATURAN PEMERINTAH 1993 MENGGUNAKAN STEERING WHEEL DAN OCULUS RIFT

Identitas Responden

Nama Lengkap : Ishardan Usia : 21 th
Pekerjaan : Mahasiswa Jenis Kelamin : L P

A. PENILAIAN TERHADAP APLIKASI

Silahkan tabel dibawah dengan menggunakan tanda (v)

SS = Sangat Setuju

S = Setuju

C = Cukup

TS = Tidak Setuju

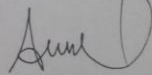
STS = Sangat Tidak Setuju

No	Parameter Antarmuka	STS	TS	C	S	SS
1	Aplikasi memiliki tampilan dan desain yang menarik				✓	
2	Aplikasi memiliki menu yang mudah digunakan					✓
3	Aplikasi Memiliki tata letak tombol yang sesuai				✓	
	Parameter Performa dan Kenyamanan					✓
4	Aplikasi nyaman untuk digunakan			✓		
	Parameter Materi Simulasi					
5	Aplikasi memberikan rasa mengendarai mobil seperti sesungguhnya					✓
6	Skenario pada aplikasi sama dengan yang sesungguhnya			✓		

B. KRITIK DAN SARAN

Fungsi rem pada aplikasi masih kurang selebihnya sudah bagus.
Kendaraan aset dengan tampilan lebih baik untuk pengembangan selanjutnya.

Surabaya, 14 Juni 2017


ISHARDAN

Gambar 0. 7 Kuesioner Responden Ketujuh

BIODATA PENULIS



Penulis lahir di kota Bukittinggi pada tanggal 20 Januari 1995, merupakan anak ke bungsu dari 3 orang bersaudara. Dalam perjalanan hidupnya penulis pernah menempuh pendidikan dasar di SD 13 Gadut Kecamatan Tilatang Kamang Agam, dilanjutkan pendidikan menengah MTsN 1 Bukittinggi, tingkat atas di SMAN 4 Bukittinggi dan terakhir di S1 Departemen Teknik Informatika Institut Teknologi Sepuluh

Nopember (ITS Surabaya) di rumpun mata kuliah Interaksi Grafika dan Seni (IGS).

Selama menjadi mahasiswa penulis pernah mengikuti pelatihan pengembangan diri mahasiswa dari tingkat Pra-Dasar, Dasar Menengah dan beberapa pelatihan lainnya. Penulis juga pernah berkecimpung di beberapa Big Event diantaranya FTIF Festival 2014, Interval ITS 2014, Schematics ITS 2014 dan 2015. Untuk bidang Organisasi penulis pernah tergabung di HMTC ITS ditahun 2014 sampai 2016, FTIF ITS tahun 2014/2015. Selain itu penulis juga pernah melaksanakan Kerja Praktik di Pusat Teknologi Informasi di PT Bank Rakyat Indonesia (PERSERO) Tbk Jakarta. Penulis dapat dihubungi lewat email fajar_ade@hotmail.com